



Universidad
Carlos III de Madrid



Universidad
Carlos III de Madrid

PROYECTO DE FIN DE GRADO

ANÁLISIS DE ENFERMEDADES A TRAVÉS DE TWITTER: TWEETFECTIONS

Autor: Guillermo Cano Yago

Titulación: Grado en ingeniería informática

Profesor: Antonio Berlanga de Jesús

Fecha: 24 de Septiembre de 2013



Índice de contenidos

Índice de contenidos	2
Índice de tablas	Error! Bookmark not defined.
Índice de ilustraciones.....	8
1. Introducción	10
1.1 Resumen.....	10
1.2 Objetivos	11
1.4 Estructura del documento.....	11
2. Estado del arte	13
2.1 Smartphone.....	13
2.2 Sistemas operativos móviles	13
2.2.1 Android.....	14
2.2.2 iOS	16
2.2.3 Elección del sistema operativo móvil.....	17
2.3 Base de datos	17
2.3.1 MySQL	17
2.3.2 PostgreSQL	18
2.3.3 Elección de SGBD.....	18
2.4 Servidor HTTP	20
2.4.2 Apache.....	21
2.4.2 IIS (Internet Information Services)	21
2.4.3 Elección del servidor	22
2.5 Scripts en el Servidor.....	22
2.5.1 PHP	23
2.5.2 ASP.NET	24
Elección del lenguaje.....	25
2.6 Twitter	25
2.7 WAMP	28
3 Análisis.....	29
3.1 Alcance	29
3.2 Alternativas de diseño.....	30
3.3 Marco regulador.....	30
3.4 Restricciones Generales	30
3.5 Entorno operacional.....	31
3.5.1 Entorno de desarrollo	31



3.5.2 Equipos	32
3.6 Análisis de Requisitos	32
3.6.1 Requisitos de Usuario.....	33
3.6.2 Requisitos de usuario de capacidad.....	34
3.6.3 Requisitos de usuario de restricción	41
3.7 Casos de Uso	42
3.7.1 Caso de uso principal.....	44
3.7.2 Caso de uso de consulta por enfermedad.....	48
3.7.3 Caso de uso de consulta por país	53
3.7.3 Caso de uso de consulta por el historial país/enfermedad.....	58
3.8 Requisitos de Software.....	63
3.8.1 Requisitos funcionales (aplicación Android)	64
3.8.2 Requisitos funcionales (Servidor).....	69
3.8.3 Requisitos no funcionales (aplicación Android)	73
3.8.4 Requisitos no funcionales (Servidor).....	77
4 Diseño.....	79
4.1 Esquema general del sistema.....	79
4.2 Aplicación móvil	80
4.2.1 Actividades.	81
4.2.2 Comunicaciones	85
4.2.3 Base datos	85
4.2.4 Interfaz de usuario	85
4.3 Servidor	100
4.3.1 Base de datos	100
4.3.2 Servidor Apache	102
4.3.2 Servicios y ficheros PHP.....	103
4.3.3 Programa en Java	104
5 Desarrollo	107
5.1 Desarrollo del servidor	107
5.1.1 Instalación de WAMP	107
5.1.2 Creación de la base de datos.....	108
5.1.3 Creación del programa buscador de tweets en Java	111
5.1.4 Comunicaciones	114
5.2 Desarrollo de la aplicación	121
5.2.1 Actividades	121



5.2.2 Layouts	128
5.3 Manifiesto	134
6 Evaluación de resultados.....	135
6.1 Tiempo de ejecución del programa en Java.....	135
6.2 Consultas desde la aplicación móvil.....	137
6.3 Coeficientes de ajuste y reparto	138
7 Presupuesto.....	141
7.1 Costes	141
8 Conclusiones y líneas futuras	144
9 Bibliografía	145



Tabla 1 RU	33
Tabla 2 RUC-01.....	34
Tabla 3 RUC-02.....	34
Tabla 4 RUC-03.....	34
Tabla 5 RUC-04.....	35
Tabla 6 RUC-05.....	35
Tabla 7 RUC-06.....	35
Tabla 8 RUC-07.....	36
Tabla 9 RUC-08.....	36
Tabla 10 RUC-09.....	36
Tabla 11 RUC-10.....	37
Tabla 12 RUC-11.....	37
Tabla 13 RUC-12.....	37
Tabla 14 RUC-13.....	38
Tabla 15 RUC-14.....	38
Tabla 16 RUC-15.....	38
Tabla 17 RUC-16.....	39
Tabla 18 RUC-17.....	39
Tabla 19 RUC-18.....	40
Tabla 20 RUC-19.....	40
Tabla 21 RUC-20.....	40
Tabla 22 RUR-01.....	41
Tabla 23 RUR-02.....	41
Tabla 24 RUR-03.....	41
Tabla 25 RUR-04.....	42
Tabla 26 CU	43
Tabla 27 CU-01.....	44
Tabla 28 CU-02.....	45
Tabla 29 CU-03.....	45
Tabla 30 CU-04.....	46
Tabla 31 CU-05.....	46
Tabla 32 CU-06.....	48
Tabla 33 CU-07.....	49
Tabla 34 CU-08.....	50
Tabla 35 CU-09.....	50
Tabla 36 CU-10.....	51
Tabla 37 CU-11.....	52
Tabla 38 CU-12.....	53
Tabla 39 CU-13.....	54
Tabla 40 CU-14.....	54
Tabla 41 CU-15.....	55
Tabla 42 CU-16.....	56
Tabla 43 CU-17.....	56
Tabla 44 CU-18.....	58
Tabla 45 CU-19.....	59
Tabla 46 CU-20.....	60
Tabla 47 CU-21.....	60
Tabla 48 CU-22.....	61



Tabla 49 CU-23.....	61
Tabla 50 CU-24.....	62
Tabla 51 RSF/RSNF.....	63
Tabla 52 RSF-01.....	64
Tabla 53 RSF-02.....	64
Tabla 54 RSF-03.....	64
Tabla 55 RSF-04.....	64
Tabla 56 RSF-05.....	65
Tabla 57 RSF-06.....	65
Tabla 58 RSF-07.....	65
Tabla 59 RSF-08.....	65
Tabla 60 RSF-09.....	66
Tabla 61 RSF-10.....	66
Tabla 62 RSF-11.....	66
Tabla 63 RSF-12.....	67
Tabla 64 RSF-13.....	67
Tabla 65 RSF-14.....	67
Tabla 66 RSF-15.....	67
Tabla 67 RSF-17.....	68
Tabla 68 RSF-17.....	68
Tabla 69 RSF-18.....	68
Tabla 70 RSF-19.....	68
Tabla 71 RSF-20.....	69
Tabla 72 RSF-21.....	69
Tabla 73 RSF-22.....	69
Tabla 74 RSF-23.....	70
Tabla 75 RSF-24.....	70
Tabla 76 RSF-25.....	70
Tabla 77 RSF-26.....	71
Tabla 78 RSF-27.....	71
Tabla 79 RSF-28.....	71
Tabla 80 RSF-29.....	71
Tabla 81 RSF-30.....	72
Tabla 82 RSF-31.....	72
Tabla 83 RSF-32.....	72
Tabla 84 RSNF-01.....	73
Tabla 85 RSNF-02.....	73
Tabla 86 RSNF-03.....	73
Tabla 87 RSNF-04.....	73
Tabla 88 RSNF-05.....	73
Tabla 89 RSNF-06.....	74
Tabla 90 RSNF-07.....	74
Tabla 91 RSNF-08.....	74
Tabla 92 RSNF-09.....	74
Tabla 93 RSNF-10.....	75
Tabla 94 RSNF-11.....	75
Tabla 95 RSNF-12.....	75
Tabla 96 RSNF-13.....	75



Tabla 97 RSNF-14	75
Tabla 98 RSNF-15	76
Tabla 99 RSNF-16	76
Tabla 100 RSNF-17	76
Tabla 101 RSNF-18	77
Tabla 102 RSNF-19	77
Tabla 103 RSNF-20	77
Tabla 104 RSNF-21	77
Tabla 105 RSNF-22	77
Tabla 106 RSNF-23	78
Tabla 107 RSNF-24	78
Tabla 108 RSNF-25	78
Tabla 109 Resultados 1	135
Tabla 110 Resultados 2	136
Tabla 111 Resultados 3	136
Tabla 112 Resultados 4	137
Tabla 113 Resultados 5	137
Tabla 114 Resultados 6	137
Tabla 115 Resultados 7	138
Tabla 116 Resultados 8	138
Tabla 117 Resultados 9	139
Tabla 118 Resultados 10	140
Tabla 119 Costes 1	141
Tabla 120 Costes 2	141
Tabla 121 Costes	142
Tabla 122 Costes	143
Tabla 123 Costes 4	143



Índice de ilustraciones

Ilustración 1 Twitter	10
Ilustración 2 Smartphones	13
Ilustración 3 Diagrama Android	15
Ilustración 4 iOS	16
Ilustración 5 Diagrama iOS.....	16
Ilustración 6 MySQL	17
Ilustración 7 PostgreSQL	18
Ilustración 8 Apache	20
Ilustración 9 IIS.....	21
Ilustración 10 PHP	24
Ilustración 11 Contenido tweets.....	25
Ilustración 12 Creación cuentas Twitter	26
Ilustración 13 API Twitter.....	27
Ilustración 14 WAMP	28
Ilustración 15 Caso de uso principal	44
Ilustración 16 Caso de uso consulta por enfermedad	48
Ilustración 17 Caso de uso consulta por país.....	53
Ilustración 18 Caso de uso consulta de historial.....	58
Ilustración 19 Esquema general del sistema.....	79
Ilustración 20 Diagrama de actividades	81
Ilustración 21 Esquema de consultas.....	83
Ilustración 22 Pantalla Android 1.....	86
Ilustración 23 Pantalla Android 2.....	87
Ilustración 24 Pantalla Android 3.....	88
Ilustración 25 Pantalla Android 4.....	89
Ilustración 26 Pantalla Android 5.....	90
Ilustración 27 Pantalla Android 6.....	91
Ilustración 28 Pantalla Android 7.....	92
Ilustración 29 Pantalla Android 8.....	93
Ilustración 30 Pantalla Android 9.....	94
Ilustración 31 Pantalla Android 10.....	95
Ilustración 32 Pantalla Android 11.....	96
Ilustración 33 Pantalla Android 12.....	97
Ilustración 34 Pantalla Android 13.....	98
Ilustración 35 Pantalla Android 14.....	99
Ilustración 36 Arquitectura WAMP	100
Ilustración 37 Tabla base de datos	101
Ilustración 38 Diagrama sistema.....	103
Ilustración 39 WAMP server	108
Ilustración 40 PHPmyAdmin 1	109
Ilustración 41 PHPmyAdmin 2	110
Ilustración 42 PHPmyAdmin 3	110
Ilustración 43 fichero PHP 1.....	115
Ilustración 44 Fichero PHP 2	116
Ilustración 45 Fichero PHP 3	117
Ilustración 46 Fichero PHP 5	118
Ilustración 47 Fichero PHP 6	119



Ilustración 48 Fichero PHP 7	120
Ilustración 49 Código 1	122
Ilustración 50 Fichero XML 1.....	128
Ilustración 51 Fichero XML 2.....	129
Ilustración 52 Fichero XML 3.....	130
Ilustración 53 Fichero XML 4.....	131
Ilustración 54 Fichero XML 5.....	131
Ilustración 55 Fichero XML 6.....	132
Ilustración 56 Fichero XML 7.....	132
Ilustración 57 Fichero XML 8.....	133
Ilustración 58 Android manifest	134
Ilustración 59 Diagrama de etapas	142

1. Introducción

1.1 Resumen

Actualmente vivimos en una era de información global. Una era en la que podemos saber qué está pasando en cualquier parte del mundo con solo unos clics de ratón o unos movimientos de dedos. Lo mejor de esta era es cómo la información ayuda a las personas a tomar mejores decisiones (siempre que la información sea objetiva) y cómo las personas demandan más información para seguir aprendiendo y sentirse más seguras.

El control de las enfermedades y plagas siempre ha sido algo que ha traído de cabeza a la sociedad puesto que afecta directamente a la seguridad de su salud. Es por ello que cada vez los estudios sobre propagación de enfermedades y la información sobre las mismas son cada vez más comunes en internet y las distintas plataformas móviles.

Por otra parte las redes sociales se están convirtiendo en parte fundamental de nuestras vidas. Al ser humano le gusta compartir experiencias y recibir información sobre sus conocidos. Es por este factor de socialización que en los últimos 10 años ha habido un crecimiento exponencial de usuarios de estas redes. Además de las citadas ventajas anteriormente, las plataformas sociales pueden servir también como otro tipo de herramientas, desde estudios de mercado “gratuitos” hasta análisis de la gramática de un idioma.

Twitter en concreto, es la red social que se utilizará en el proyecto. Esta red social sirve a las personas para

expresar su estado de ánimo pensamientos o compartir distintos contenidos. La faceta que nos interesa en el presente proyecto son aquellos tweets de personas que expresan su estado de salud cuando están



Ilustración 1 Twitter

afectadas por enfermedades. Y, mediante la localización de estos tweets crear un “mapa” para dichas enfermedades.

El proyecto que se expone a continuación propone una herramienta a través de la cual un usuario pueda acceder a la información las enfermedades que afectan a distintas personas en varios países. Pudiendo así programar viajes o simplemente conocer qué enfermedades se encuentran más propagadas en su país. Todo ello a través de una interfaz cuidada y simple en forma de aplicación móvil y un servidor propio que se encargará de toda la labor de conexión y búsqueda de datos.



Los principales usos del sistema serían permitir a las personas acceder a información sobre el estado de ciertas enfermedades en diversos países y ayudar a las personas a planear sus vacaciones si tienen alguna sensibilidad especial hacia una enfermedad que pudiera ser determinante para su disfrute.

1.2 Objetivos

El objetivo principal del proyecto es el de desarrollar un sistema que mediante la red social Twitter permita a un usuario conocer la propagación de ciertas enfermedades en determinados países.

Para alcanzar los objetivos, el sistema se compondrá de dos subsistemas:

- Una aplicación móvil a través de la cual el usuario pueda acceder a la información recopilada.
- Un servidor que se encargue de atender las peticiones de la aplicación móvil y además recopile la información que se va a servir.

Y por último, será necesario cumplir con los siguientes requisitos:

- Conocer el lenguaje de programación Android
- Conocer el lenguaje de programación Java.
- Conocer el lenguaje MySQL
- Conocer el lenguaje de programación PHP
- Conocer el funcionamiento de algunas partes de la API de Twitter.
- Desarrollar una aplicación móvil en Android
- Implementar una base de datos MySQL
- Implementar un servidor Apache
- Desarrollar un programa en Java
- Diseñar un sistema de captación y tratamiento de tweets
- Diseñar las conexiones entre los distintos elementos de los subsistemas

1.4 Estructura del documento

En este apartado se exponen las distintas partes del documento y se explica brevemente en qué consisten.

- **Introducción:** Breve resumen e idea general del proyecto incluyendo los objetivos a conseguir y la presente estructura del documento.
- **Estado del arte:** Explicación de las tecnologías escogidas para el proyecto así como las alternativas que se han desechado. Se analizarán la tecnología utilizada para el servidor, para la creación de la aplicación, la base de datos y las conexiones entre los elementos.



- **Análisis:** Se exponen las necesidades del sistema así como un análisis de los casos de uso y los requisitos de usuario y de software es decir todo aquello que debe ser satisfecho por el sistema. También se incluyen las alternativas de diseño.
- **Diseño:** En esta parte se explica el diseño de la aplicación y su arquitectura así como el funcionamiento de los distintos componentes y su función el sistema. Se define el modelo de la base datos, la interfaz de la aplicación, las consultas a la misma, el programa utilizado para buscar tweets...
- **Desarrollo:** En esta sección se detallan las funcionalidades internas del sistema. Se explicará el desarrollo del servidor y todos sus componentes y de la aplicación. Además se incluirán los estudios realizados para el ajuste de los datos recopilados.
- **Evaluación de resultados:** Aquí se expone un análisis de los resultados obtenidos y su fiabilidad además se mide la escalabilidad del sistema y los tiempos de respuesta. Se incluyen también los datos que respaldan los coeficientes de ajuste creados para el sistema.
- **Presupuesto** En esta parte se mostrará el coste del desempeño del proyecto.
- **Conclusiones:** Breve resumen de lo conseguido, las ideas extraídas de su ejecución y trabajos y líneas futuras.
- **Bibliografía:** Fuentes utilizadas durante la redacción de la memoria y del proyecto.

2. Estado del arte

El proyecto consta de varias tecnologías en su desarrollo. Para entender mejor su funcionamiento es conveniente explicar el porqué de la elección dichas tecnologías, citadas a continuación. Además se hará una comparación con tecnologías similares que se han desechado.

2.1 Smartphone

Los Smartphone o teléfonos inteligentes son teléfonos móviles contruidos sobre una plataforma informática con una capacidad superior para almacenar datos y realizar actividades similares a las de un ordenador. La mayoría de estos smartphones son táctiles y por lo general multitarea.

Todo esto ha hecho posible uno de los cambios más notables producidos en el siglo 21: la inclusión en la vida cotidiana de los smartphones. La propagación de los smartphones en todo el mundo ha permitido que la comunicación instantánea sea posible en cualquier lugar. Y, como no podía ser de otra forma, las redes sociales desarrollan muchas aplicaciones para hacer atractivo el acceso a través de smartphones y así captar más clientes y usuarios.



Ilustración 2 Smartphones

Actualmente existen varias marcas que crean smartphones algunas incluso con su propio sistema operativo. Las más destacadas son: Sony, Apple, Samsung, Nokia, LG, Motorola, HTC, y BlackBerry. En el caso de este proyecto el dispositivo utilizado será un Samsung Galaxy SII pero no queda restringido a este dispositivo. La

restricción de aquellos dispositivos en los que se puede utilizar la da el Sistema Operativo del móvil sobre los cuales se hablan a continuación.

2.2 Sistemas operativos móviles

Un sistema operativo móvil es un sistema operativo que controla un dispositivo móvil. Los que nos interesan en esta sección son los de los smartphones. Estos están orientados a la conectividad inalámbrica los formatos multimedia, las redes sociales y las distintas maneras de introducir información en ellos y comunicarla.

Las principales capas de los sistemas operativos móviles son:

- Kernel: es el núcleo y proporciona acceso a los distintos elementos del hardware del dispositivo.



- Middleware: es el conjunto de módulos que hacen posible la existencia de aplicaciones para móviles y ofrece servicios como el motor de mensajería, codecs multimedia, interpretes web...
- Entorno de ejecución: consiste en un gesto de aplicaciones abiertas y programables por parte de los desarrolladores para facilitar la creación de software.
- Interfaz de usuario las interfaces facilitan la interacción con el usuario y el diseño visual de la aplicación. Incluye servicios gráficos tales como botones pantallas o listas.

La cuota de mercado de los sistemas operativos de los smartphones en el primer trimestre de 2013 era:

1. Android 75,0% (En países como España las diferencias son más significativas, donde Android tiene el 87% de la cuota de mercado.¹)
2. iOS 17.3%
3. Windows Phone y Windows Mobile 3,2%
4. BlackBerry OS 2,9%
5. Linux u otros 1,0%
6. Symbian OS 0,6%

Dado que aproximadamente el 95% del mercado lo tienen Android e iOS en esta sección se evaluarán estos dos.

2.2.1 Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

Las versiones de Android hasta la fecha han sido: Apple Pie (v.1.0), Banana Bread (v.1.1), Cupcake (v.1.5), Donut (v.1.6), Éclair (v.2.0), Froyo (v.2.2), Gingerbread (v.2.3), Honeycomb (v.3.0), Ice Cream Sandwich (v.4.0), Jelly Bean (v.4.1).

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma

muy sencilla en un lenguaje de programación muy conocido como es Java.

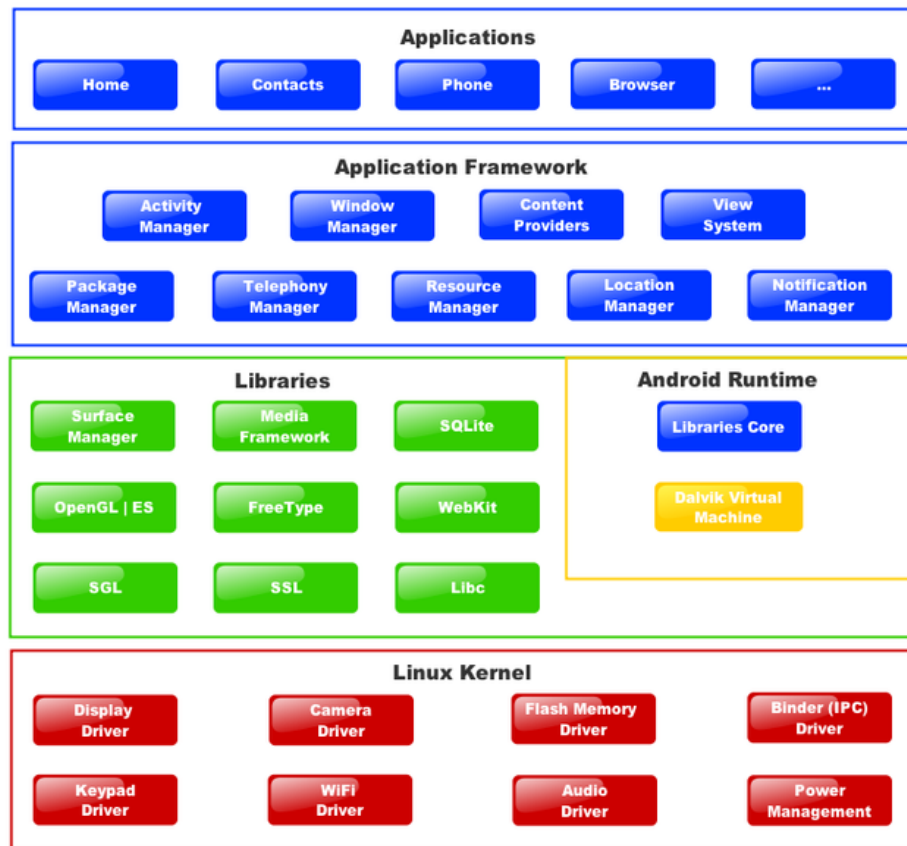


Ilustración 3 Diagrama Android

Una de las mejores características de este sistema operativo es que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada. Y esto lo hace muy popular entre fabricantes y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos.

Cualquiera puede bajarse el código fuente, inspeccionarlo, compilarlo e incluso cambiarlo. Esto da una seguridad a los usuarios, ya que algo que es abierto permite detectar fallos más rápidamente. Y también a los fabricantes, pues pueden adaptar mejor el sistema operativo a los terminales.

De cara al desarrollo de una aplicación existen tanto ventajas como inconvenientes. Por un lado existen gran variedad de librerías entre las que el programador puede elegir la que más convenga para su aplicación. Además la comunidad de desarrolladores es enorme por lo que siempre se proporciona ayuda a quien la necesita y casi todas las dudas que uno pueda tener están resueltas porque alguien las tuvo antes. Por otro lado la parte negativa de Android es que, dado la cantidad de dispositivos que lo utilizan hay que adaptar el diseño de las aplicaciones para que puedan ser utilizadas por todos, o al menos la gran mayoría.

2.2.2 iOS

iOS es un sistema operativo móvil de la empresa Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple, Inc. no permite la instalación de iOS en hardware de terceros.

La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles. Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).



Ilustración 4 iOS

Las aplicaciones deben ser escritas y compiladas específicamente para la arquitectura ARM, por lo que las desarrolladas para Mac OS X no pueden ser usadas en iOS. Además debe pagarse por el kit de desarrollo y se debe contar con un Mac o emulador para desarrollarlas. En la siguiente imagen se puede apreciar la arquitectura de iOS:

iPhone Architecture



Ilustración 5 Diagrama iOS



2.2.3 Elección del sistema operativo móvil

En este proyecto se ha decidido utilizar Android por dos razones muy simples, la primera es que es gratuito con lo que los costes se reducen y la segunda es que es un lenguaje utilizado a lo largo de la carrera con lo que la velocidad para programar con él ha sido muy rápida lo que ha permitido incluir algunas funcionalidades que con otro lenguaje habrían resultado más complicadas y habrían conllevado más tiempo.

2.3 Base de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados para su posterior uso. Existen una serie de programas denominados sistemas gestores de bases de datos. Estos permiten almacenar y acceder a los datos de manera rápida. La base de datos a desarrollar será una base de datos dinámica con contenido cambiante y será del tipo relacional. Como el tipo de base de datos ya está decidida y para este tipo de proyecto no se ha contemplado otra alternativa, en esta sección se decidirá qué sistema gestor de base de datos se va a utilizar.

Los SGBD más utilizados son:

- MySQL
- Postgress
- Access
- Oracle

Para el proyecto sólo se han contemplado las alternativas gratuitas (las dos primeras opciones de la lista) por lo que la elección se hará entre MySQL y Postgress

2.3.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.¹MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Existen varias interfaces de programación de aplicaciones que permiten, a



Ilustración 6 MySQL



aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una interfaz de programación de aplicaciones específica.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL funciona sobre múltiples plataformas, incluyendo:

AIX, BSD, FreeBSD, HP-UX, Kurisu OS, GNU/Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServe, SCO UnixWare, Tru64, eBD, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8 y Windows Server (2000, 2003, 2008 y 2012) y OpenVMS

2.3.2 PostgreSQL

PostgreSQL es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*.

Algunas de sus ventajas frente a otros gestores son Seguridad en terminos generales -Integridad en BD: restricciones en el dominio -Integridad referencial - Afirmaciones (Assertions) -Disparadores (Tiggers) -Autorizaciones -Conexión a DBMS -Transacciones y respaldos



Ilustración 7 PostgreSQL

2.3.3 Elección de SGBD

Lo bueno de Mysql de cara a ser utilizado por un desarrollador de APPS de Android es que su curva de aprendizaje es muy rápida, además existen infinidad de herramientas e información



para su manejo por lo que se obtiene mucha rapidez de cara a configurar un entorno. Además a nivel usuario es gratuito y de cara a este proyecto encajaba perfectamente con php, apache y Android. PostgreSQL por su parte no está tan extendida y al no conocer el sistema con anterioridad se ha tomado la decisión de elegir MySQL como sistema gestor.

2.4 Servidor HTTP

Un servidor web o servidor http es un programa informático que procesa una aplicación del lado del servidor y realiza conexiones bidireccionales o unidireccionales síncrona o asíncronas con el cliente generado. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.

Un servidor web opera mediante el protocolo HTTP por lo que las peticiones se realizan mediante GET. El segundo tipo de petición más utilizado es la petición POST y se utiliza para enviar datos al servidor.

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente

Algunos de los servidores más importantes son Nginx, Apache, IIS o Tomcat. Los servidores que se han contemplado para el desarrollo del proyecto son IIS y Apache por su extensión en internet y facilidad de uso

2.4.1 Apache

Visión general e historia

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1² y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3,

pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, *a patchy server* (un servidor "parcheado") suena igual que *Apache Server*.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los

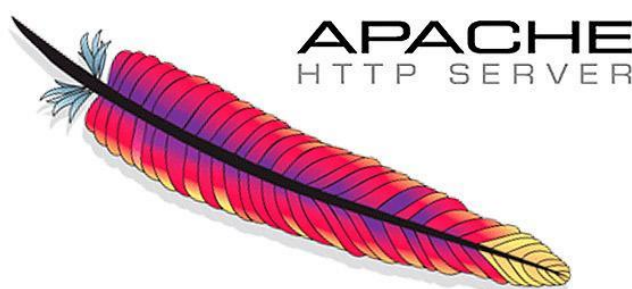


Ilustración 8 Apache

sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

Características principales

- Licencia GPL gratuita.
- Multiplataforma
- Gran cantidad de documentación y dudas resueltas
- Rápido
- No es fácil de configurar
- Sus actualizaciones no son regulares
- Interfaz sencilla y amigable
- Modular y extensible.
- Puede conectarse directamente a una base de datos
- Gran estabilidad
- Excelente integración con PHP y MySQL

2.4.2 IIS (Internet Information Services)

Visión general e historia

IIS es un servidor web y conjunto de servicios para el sistema operativo Microsoft Windows. Se fue integrando poco a poco con algunos sistemas operativos de Microsoft para ofrecer distintos servicios como en Windows 2000 o Windows Server 2003. Este servicio convierte a un ordenador en un servidor web de internet, es decir que se pueden publicar páginas web o proporcionar servicios en internet.



Ilustración 9 IIS

IIS fue inicialmente lanzado como un conjunto de servicios basados en Internet para Windows NT 3.51. IIS 2.0 siguió agregando soporte para el sistema operativo Windows NT 4.0 y IIS 3.0 introdujo las Active Server Pages, una tecnología descripting dinámico.

IIS 4.0 eliminó el soporte para el protocolo Gopher y fue puesto con Windows NT como un CD-ROM de "Paquete Opcional" separado.

La versión de IIS 7.5 corresponde a Windows Server 2008, e IIS 5.1 a Windows XP Professional. La versión IIS 5.1, para Windows XP, es una versión compacta del IIS que soporta sólo 10 conexiones simultáneas y sólo un sitio web. IIS 6.0 ha agregado soporte para IPv6.



Windows Vista viene con IIS 7.0 preinstalado. No limitará el número de conexiones permitidas pero limitará el flujo de tareas basándose en las solicitudes activas concurrentes, mejorando el uso y el rendimiento en escenarios punto-a-punto

Características

- Es fiable y seguro
- Es fácil de usar
- Código fuente propietario
- No tiene coste adicional si tienes Windows
- Desarrollado por Microsoft
- Se puede administrar desde Internet
- Licencia de pago
- No es multiplataforma

2.4.3 Elección del servidor

Dado que lo que se busca es un software multiplataforma, fácil de configurar y con una buena integración con base de datos se ha decidido utilizar Apache. Además Apache se integra perfectamente con PHP y MySQL que son los otros componentes elegidos para conformar el sistema. Además el conocimiento del funcionamiento de Apache y la facilidad para encontrar recursos y ayudas en Internet harán que la configuración y utilización de este sea mucho más rápida y sencilla.

2.5 Scripts en el Servidor

La programación en el lado del servidor es un tipo de programación que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor para generar respuestas de manera dinámica.

Los primeros servidores web permitían visualizar exclusivamente información estática. Esto presentó pronto una limitación; sobre todo desde el momento en el que la actividad publicitaria y comercial comenzó a concentrarse también en Internet. La primera solución técnica realizada fue la posibilidad de que el servidor web ejecutase programas residentes en la máquina de servicio. Esta tecnología, conocida como *Common Gateway Interface* (CGI) permitía lanzar programas escritos principalmente en C o Perl.

Si bien la tecnología CGI resolvía el problema de la presentación exclusiva de información estática, al mismo tiempo presentaba dos limitaciones importantes: el problema de seguridad que podía representar el hecho de que mediante una petición se pudiesen ejecutar programas indeseados en el servidor y la carga del servidor (si una página que lanzaba un programa era llamada desde 100 clientes simultáneamente, el servidor ejecutaba 100 procesos, uno por cada cliente que solicitaba dicha página).



Para resolver estos problemas, se buscó desarrollar una tecnología que permitiera ejecutar, en un único proceso del servidor, todos los pedidos de ejecución de código sin importar la cantidad de clientes que se conectaban concurrentemente. Así surgieron los denominados *servlets*, basados en la tecnología Java de Sun Microsystems, y los filtros ISAPI de Microsoft. Éstos permitían ejecutar código en un único proceso externo que gestionaba todas las llamadas realizadas por el servidor web, impidiendo al mismo tiempo que el servidor web pueda ejecutar programas del sistema operativo.

Los lenguajes de script más utilizados son PHP, ASP y Ruby. A continuación se analizarán PHP y ASP dado que eran las alternativas más viables dados los conocimientos del alumno y la integración con los demás componentes.

2.5.1 PHP

Visión general e historia

Fue originalmente diseñado en Perl, con base en la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés-canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vitae y guardar ciertos datos, como la cantidad de tráfico que su página web recibía. El 8 de junio de 1995 fue publicado "Personal Home Page Tools" después de que Lerdorf lo combinara con su propio *Form Interpreter* para crear PHP/FI.

Dos programadores israelíes del Technion, Zeev Suraski y Andi Gutmans, reescribieron el analizador sintáctico (*parser* en inglés) en el año 1997 y crearon la base del PHP3, cambiando el nombre del lenguaje por *PHP: Hypertext Preprocessor*. Inmediatamente comenzaron experimentaciones públicas de PHP3 y fue publicado oficialmente en junio de 1998. Para 1999, Suraski y Gutmans reescribieron el código de PHP, produciendo lo que hoy se conoce como motor Zend. También fundaron Zend Technologies en Ramat Gan, Israel.

En mayo de 2000 PHP 4 fue lanzado bajo el poder del motor Zend Engine 1.0. El día 13 de julio de 2007 se anunció la suspensión del soporte y desarrollo de la versión 4 de PHP, a pesar de lo anunciado se ha liberado una nueva versión con mejoras de seguridad, la 4.4.8 publicada el 13 de enero del 2008 y posteriormente la versión 4.4.9 publicada el 7 de agosto de 2008. Según esta noticia¹³ se le dio soporte a fallos críticos hasta el 9 de agosto de 2008.

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine 2.0



PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún coste.



Ilustración 10 PHP

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Características

- Puede ser desplegado en la mayoría de servidores web
- Puede ser desplegado en casi todo los sistemas operativos
- Es gratuito
- Está instalado en más de 20 millones de sitios web
- Es el módulo de Apache más popular para ese servidor web
- Permite la conexión a varios tipos de servidores de bases de datos como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

2.5.2 ASP.NET

Visión general e historia

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Cualquier persona que esté familiarizada con el desarrollo de aplicaciones web sabrá que el desarrollo web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de



etiquetas, un gran uso de lenguajes de *script* y plataformas deservidor. Para el programador de nivel intermedio, el conocimiento y las habilidades que se necesitan para desarrollar aplicaciones web requieren de un mayor conocimiento tanto de lenguajes de programación, etiquetado y formato, como de diversas tecnologías de software relativas al desarrollo distribuido y concurrente, de las que son necesarias en el desarrollo tradicional de aplicaciones.

Características

- Su principal herramienta son los web forms
- Permite la creación de componentes reutilizables.
- Las aplicaciones son alojadas en un servidor web y se tiene acceso a ellas mediante HTTP
- Se debe implementar su propia administración para el estado de la conexión
- Requiere un tiempo de programación grande
- Su licencia no es gratuita
- Curva de aprendizaje extensa
- Poca cantidad de librerías gratuitas

Elección del lenguaje

Se ha elegido PHP como lenguaje de programación para los scripts de servicio del servidor. Esto se debe a que es gratuito, se integra perfectamente con MySQL y Apache y además es sencillo de aprender a utilizar ya que tiene muchos parecidos con C y Java.

2.6 Twitter

Visión general e historia

Twitter es un servido de microblogging. Se creó en 2006 en San Francisco como una start-up y al principio se usaba internamente pero posteriormente fue lanzada al público ese mismo año.

El nombre original del servicio de mensajería iba a ser "twittr", inspiración de Glass en alusión a "Flickr". La versión definitiva se lanzó el 15 de julio de 2006. En Octubre de ese año, Stone, Williams, Dorsey y otros formaron Obvious Corporation, adquiriendo los derechos sobre Odeo, luego de que los inversores originales perdieran su interés ante el fracaso inicial del proyecto. Desde esa fecha, Twitter ha ido ganando adeptos rápidamente. En marzo de 2007, ganó el premio South by Southwest Web Award en la categoría de blog. Y en abril, la nueva compañía Twitter se independizó de su gestora.

A principios de 2008, el equipo de Twitter estaba compuesto por 18 personas. Al año siguiente, Twitter había multiplicado su plantilla por cuatro y seguía creciendo.

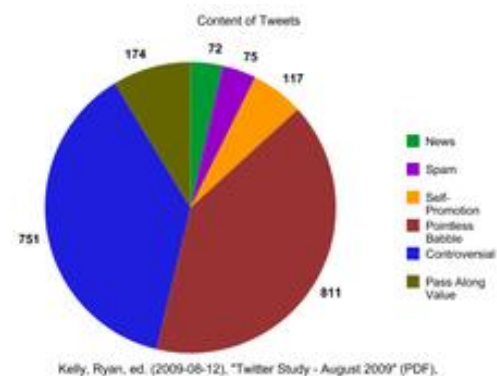


Ilustración 11 Contenido tweets

Aunque Twitter usó durante un corto período de tiempo los servicios de publicidad como AdSense de Google, la compañía decidió descartar los ingresos por publicidad hasta que aumentara el número de usuarios, y se financió mientras tanto con inversiones de empresas de capital riesgo. En septiembre de 2009, Twitter anunció cambios a las condiciones de servicio, dejando abierta la posibilidad de incluir publicidad en sus servicios.

Twitter ha estado entre los diez sitios web más visitados del mundo según el sitio web Alexa.²¹ Las estimaciones diarias de un usuario pueden variar ya que la compañía no publica estadísticas sobre las cuentas activas. A febrero de 2009 el blog Compete.com calificó a Twitter como la tercera red social más visitada basándose en el contenido 6 millones de visitantes únicos y 55 millones de visitas mensuales.²² En marzo de 2009, Nielsen.com calificó a Twitter como una de las webs con mayor crecimiento estimado para febrero de 2009. Twitter crecía anualmente un 1382%, pasando de 475.000 visitantes únicos en febrero de 2008 a 7 millones en febrero de 2009. Fue seguido por Zimbio con un aumento del 240% y Facebook con un 228%.²³ Sin embargo, Twitter tiene una tasa de retención del usuario de un 40%.

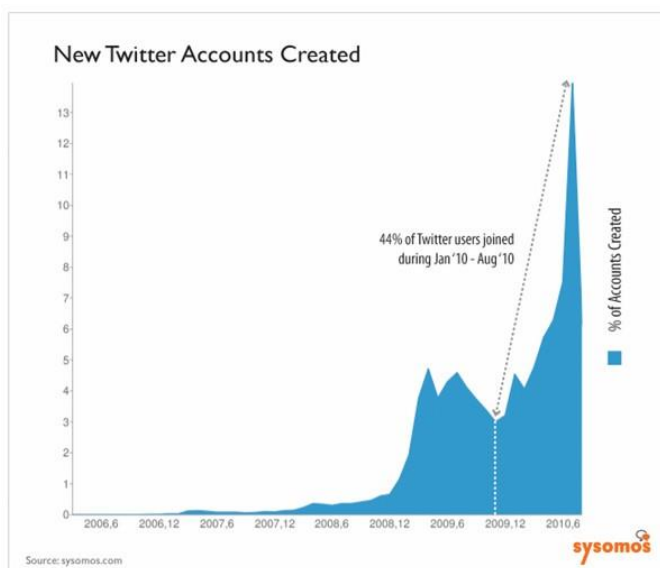


Ilustración 12 Creación cuentas Twitter

La interfaz está desarrollada en Ruby on Rails pero los mensajes se guardan en el servidor mediante el lenguaje Scala. Twitter liberó una API para que los desarrolladores pudieran integrar Twitter en sus servicios y aplicaciones. Más del 50% del tráfico de Twitter proviene de su API que es lo que usa este proyecto y que se analiza a continuación.

API de twitter

La API de twitter permite el acceso a gran parte de la información de Twitter. Twitter ofrece varias herramientas mediante las cuales los programadores pueden usar Twitter desde sus aplicaciones. Hasta hace unos meses se utilizaba la versión 1 pero de uno meses a esta parte se ha cambiado a la v1.1 restringiendo el acceso al servicio. Por ello para realizar el proyecto se ha tenido que desarrollar un mecanismo que permita recoger la mayor cantidad de información diaria proporcionada por twitter.

La siguiente ilustración muestras las principales características de la API de Twitter versión 1.1.

Lo que siempre quiso saber del API de Twitter y nunca se atrevió a preguntar

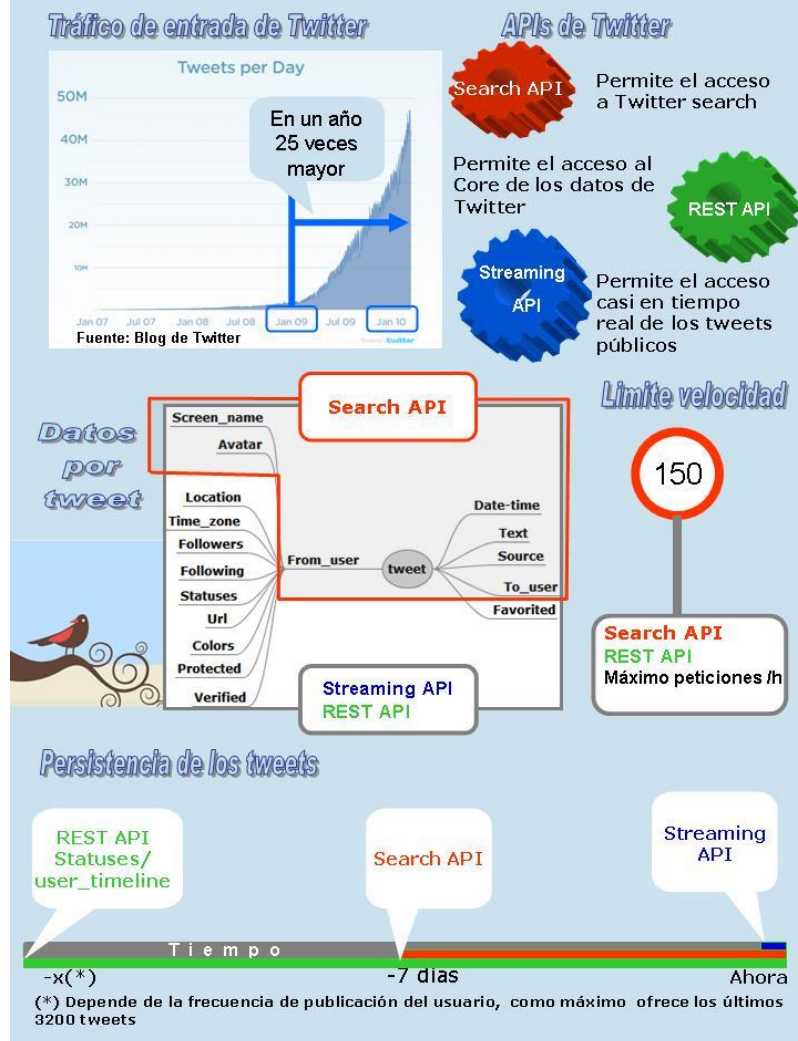


Ilustración 13 API Twitter

Como se puede observar se distinguen claramente 3 APIs:

REST API: ofrece a los desarrolladores el acceso al core de los datos de Twitter. Todas las operaciones que se pueden hacer vía web son posibles realizarlas desde el API. Dependiendo de la operación requiere o no autenticación, con el mismo criterio que en el acceso web. Soporta los formatos: xml, json, rss, atom.

Search API: suministra los tweets con una profundidad en el tiempo de 7 días que se ajustan a la query solicitada. Es posible filtrar por, cliente utilizado, lenguaje y localización. No requiere autenticación y los tweets se obtienen en formato json o atom.

Streaming API: proporciona un subset de tweets en casi tiempo real. Se establece una conexión permanente por usuario con los servidores de Twitter y mediante una petición http se recibe un flujo continuo de tweets en formato json. Se puede obtener una muestra aleatoria (statuses/sample), un filtrado (statuses/filter) por palabras claves o por usuarios. Sin

embargo, los métodos más interesantes cómo obtener todo el caudal de tweets (statuses/firehose) o sólo los tweets que tienen enlaces (statuses/links) o los tweets con retweets (statuses/retweet)

Pero a partir junio de 2013 la política de acceso a Twitter mediante aplicaciones de terceros ha cambiado y pasa por pagar para acceder a X volumen de datos. Por lo tanto para el desarrollo del buscador de tweets se han realizado una serie de variaciones sobre la idea inicial para poder realizar las búsquedas. Estas variaciones se explicarán en la sección de Diseño y posteriormente en la sección de “Desarrollo”.

2.7 WAMP

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo;
- Apache, como servidor web;
- MySQL, como gestor de bases de datos;
- PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un WAMP permite servir páginas HTML a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web. Es una herramienta perfecta para este proyecto por lo que, en vez de instalar cada componente por su parte se ha decidido utilizar WAMP dado que es gratuita y además proporciona una instalación muy sencilla y fácil de configurar.



Ilustración 14 WAMP



3 Análisis

En esta sección se incluye un análisis detallado de las capacidades y características del proyecto desarrollado. Además se incluirá una descripción general de su funcionamiento interno para dar sentido a las posteriores secciones.

3.1 Alcance

El sistema a desarrollar se compondrá de dos partes: una parte de aplicación en Android para mostrar al usuario la información consultada y otra parte de servidor en el que se almacenarán los datos recogidos y será la parte que servirá la información a la aplicación de usuario. Son partes totalmente diferenciadas y que solo coinciden en la solicitud y respuesta de unos datos determinados.

Por un lado la aplicación deberá ser capaz de proporcionar al usuario información recogida de twitter de una manera clara y concisa a través de tres tipos distintos de gráficos: gráfico de barras, gráfico de tarta y gráfico de puntos. El usuario podrá realizar tres acciones y para cada acción podrá elegir entre dos modalidades de recogida de datos: una en la que sólo se usen los tweets con localización y otra en la que se incluyen también aquellos que no tienen localización. Una de las acciones será elegir un país, una fecha de inicio, una fecha de fin y podrá ver en un gráfico de tartas los porcentajes de qué enfermedades están más extendidas. Otra acción permitirá al usuario elegir una enfermedad, una fecha de inicio y una fecha de fin y tendrá acceso a una grafico de barras que listarán los países en función del índice de enfermos de esa enfermedad que se han registrado. Y por último, el usuario podrá elegir un país, una enfermedad, una fecha de inicio y una fecha de fin y observará en un gráfico de puntos la evolución en esas fechas de la enfermedad seleccionada en el país seleccionado.

Por otro lado el sistema de servidor se compondrá de 4 elementos principales:

- **Programa en Java:** Es el corazón del sistema, este programa es un motor de búsqueda que utiliza la API de Twitter para recopilar datos. Se ejecuta en principio por el administrador del sistema una vez al día y almacena los datos en la base de datos mediante consultas php y conexión con un servidor Apache en red local que contiene una base de datos MySQL.
- **Servidor Apache:** Se encarga de almacenar la base de datos y proporcionar un acceso remoto a esta. Además ejecuta las sentencias mysql de consulta actualización borrado e inserción escritas en php.
- **Scripts en PHP:** una serie de scripts que contienen métodos para ejecutar acciones sobre la base de datos del servidor. Estos scripts son usados tanto por el programa en



Java como por la aplicación en Android.

- **Base de datos MySQL:** El almacén de la información. Se encarga de servir datos al servidor quien a su vez devuelve estos datos a la aplicación en Android y al programa en Java. Sigue un modelo relacional que se explicará más adelante.

3.2 Alternativas de diseño

En esta sección se detallan algunas alternativas de diseño que se han encontrado durante o tras el desarrollo del proyecto y que no se han implementado por falta de recursos o de tiempo:

- Realizar la parte de servidor mediante el uso del servicio de servidor GAE que ofrece Google. Además se podría haber realizado de manera gratuita ya que el límite de volumen que se tiene antes de tener que pagar, es más que suficiente para la realización del proyecto.
- Incluir una lista detallada de principales ciudades y pueblos de cada país para que el índice de localizaciones positivas fuese mayor que el actual.
- Utilizar robots de rastreos de tweets (previo pago) para rastrear las IDs de los usuarios que generan los tweets capturados y así tener localizaciones más exactas en vez de utilizar localizaciones fijas en el sistema.

3.3 Marco regulador

El sistema accede y almacena información de tweets de usuarios de Twitter por lo tanto es conveniente ver en qué puede afectar al desarrollo del sistema y su uso la Ley orgánica 15/1999 de Protección de Datos de Carácter Personal.

Esta ley tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.

El sistema accede a los tweets pero no almacena ni el contenido y la información personal del usuario. Sólo almacena la localización durante unos minutos pero no se relaciona con el usuario por lo tanto dado el carácter impersonal de los datos guardados y el no almacenamiento de datos personales la Ley de Protección de Datos de Carácter Personal no afecta a ninguna parte de la aplicación

3.4 Restricciones Generales



En función de lo estipulado entre el tutor y el alumno y la ley vigente en este tipo de sistemas las restricciones generales del proyecto son las siguientes:

- El sistema no almacenará información personal de los usuarios (esto se aplica a las búsquedas que realiza, su localización, datos del dispositivo...)
- La aplicación del usuario funcionará en Android.
- La aplicación de usuario ejecutará un pequeño tutorial la primera vez que se abra
- La aplicación del usuario tendrá una interfaz sencilla clara y accesible.
- La información mostrada en la aplicación de usuario será clara.
- El sistema será robusto y no tendrá fallos de compilación ni de funcionamiento.
- Los datos recogidos de tweets sin localización deberán ser correctamente ponderados.
- La base de datos se desarrollará en MySQL.
- La conexión entre aplicación y servidor se realizará mediante php
- El servidor será Apache
- El programa en Java del servidor se ejecutará todos los días
- El servidor responderá en un tiempo razonable y se contemplará una posible no respuesta.

3.5 Entorno operacional

Como se ha mencionado anteriormente el sistema está compuesto de dos partes. Por lo tanto existen dos entornos operacionales:

-Aplicación Android: Una versión del sistema operativo igual o superior a Icecream Sandwich (v.4.0). Una cantidad de memoria RAM no inferior a 400 Mb (no es mucha ya que el sistema no necesita capacidad de cómputo). Conexión a internet vía datos o vía WI-FI para poder acceder al servidor remoto.

-Servidor: Sistema operativo Windows 7, conexión constante a internet para responder a las consultas en cualquier momento. Memoria RAM no inferior a 4GB y un procesador superior a 2,5 Ghz ya que es necesario que la ejecución del programa en java se realice medianamente rápido. Java instalado en una versión igual o superior a la v7 update 40 y Visual C++ 2010 para la correcta ejecución del servidor apache. Además se deberá configurar un puerto del router para permitir conexiones entrantes y salientes del servidor.

3.5.1 Entorno de desarrollo

Para la creación gestión y mantenimiento del servidor y la base de datos se utilizará WAMP en su versión 2.4. WAMP es un paquete de software que instala automáticamente las últimas versiones de Apache MySQL y PHP con lo que se ahorra mucho tiempo. Además su interfaz es muy sencilla y permite gestionar la base de datos y el servidor de manera intuitiva y rápida.



Eclipse se utilizará tanto para el desarrollo del motor de búsqueda en java como para la aplicación móvil en su versión utilizando un plugin de Android llamado ADT (android development tools) en su versión 22.0.5. Se utilizará la versión 4.3 de Eclipse ya que es la última y la que más funcionalidades y compatibilidades tiene. Además Eclipse es un entorno de desarrollo muy extendido por lo que de cara a la resolución de problemas se hará de una manera más rápida que utilizando un entorno menos conocido.

3.5.2 Equipos

En esta sección se detallan los equipos utilizados para el desarrollo y la administración del sistema.

Ordenador de sobremesa:

- Procesador: Intel Core i7-3600K 3.4GHz.
- Sistema Operativo: Windows 7 Professional 64 bits.
- Memoria RAM: 8GB DDR3.
- Capacidad del disco duro: 1TB.

Dispositivo móvil Samsung Galaxy SII:

- Modelo: GT-I9100.
- Versión del sistema operativo: 4.1.2 Jelly Bean (API 16). [3].
- Procesador: Dual-Core 1.2 GHz Cortex-A9. [4].
- RAM: 1GB.
- Versión del Kernel: 3.0.31

3.6 Análisis de Requisitos

El análisis de requisitos es algo fundamental en el desarrollo de cualquier proyecto. Gracias a estos requisitos se pueden tener unos objetivos firmes y claros durante la creación de programas o aplicaciones. Además sirven para dejar claras las necesidades de los clientes y aquello que los diseñadores y programadores pueden ofrecerles. Dado que este caso es algo particular puesto que no hay un cliente claro, se tomará como cliente tanto al tutor como al alumno y se llevarán a cabo los requisitos que ambos establezcan.

Además como este proyecto tiene dos partes diferenciadas (aplicación de usuario y servidor) los requisitos también se diferenciarán en función de la parte a la que pertenezcan. En primer lugar se analizarán los requisitos de usuario y de software de la aplicación y posteriormente los del sistema servidor. De los requisitos de usuario se obtendrán los casos de uso, y, posteriormente los requisitos de software.



3.6.1 Requisitos de Usuario

Los requisitos de usuario son aquellos que especifican lo que el “cliente” necesita y se pueden distinguir dos tipos: de restricción y de capacidad. Para su mejor comprensión se ha decidido crear una serie de tablas en las que se incluye la información de cada requisito. Estas tablas contendrán los siguientes campos:

- **Requisito:** Es el identificador del requisito. Es exclusivo y auto descriptivo para que de un vistazo rápido pueda identificarse. Se compone de siglas que se refieren al tipo de requisito y número para que, en el caso de dos requisitos con las mismas siglas, puedan distinguirse.
- **Descripción:** Breve información sobre qué debería poder hacer el usuario o que no debería poder hacer
- **Fuente:** De dónde ha salido el requisito (alumno, tutor o decidido conjuntamente).
- **Fit Criterion:** Conjunto de condiciones que deben cumplirse para el correcto desarrollo de la funcionalidad descrita.
- **Prioridad:** La prioridad de un requisito frente al resto. Aquellos con mayor prioridad serán implementados antes que los que tienen una prioridad menor los posibles valores serán Alta, Media y Baja.
- **Importancia:** La necesidad de implementar un requisito y su impacto en el sistema. Los posibles valores serán 1 (totalmente necesario) 2 (es necesario pero no influye en el funcionamiento del programa/aplicación) 3 (no es necesario ni influye en el funcionamiento del programa/aplicación)

Tabla 1 RU

	RU%-%%
Descripción	
Fuente	
Fit Criterion	
Prioridad	
Importancia	



3.6.2 Requisitos de usuario de capacidad

Estos requisitos definen todas aquellas acciones que, a través del sistema, puede realizar el usuario.

Tabla 2 RUC-01

Requisito	RUC-01
Descripción	El usuario tendrá acceso a un tutorial la primera vez que se ejecute la aplicación
Fuente	Tutor
Fit Criterion	Ejecutar la aplicación por primera vez
Prioridad	Media
Importancia	2

Tabla 3 RUC-02

Requisito	RUC-02
Descripción	El usuario podrá acceder a la consulta por enfermedad
Fuente	Decisión conjunta
Fit Criterion	Pulsar el botón "Consulta por enfermedad"
Prioridad	Alta
Importancia	1

Tabla 4 RUC-03

Requisito	RUC-03
Descripción	El usuario podrá acceder a la consulta por país
Fuente	Decisión conjunta
Fit Criterion	Pulsar el botón "Consulta por país"
Prioridad	Alta
Importancia	1



Tabla 5 RUC-04

Requisito	RUC-04
Descripción	El usuario podrá acceder a la consulta del historial país/enfermedad
Fuente	Decisión conjunta
Fit Criterion	Pulsar el botón “Consultar historial País/Enfermedad”
Prioridad	Alta
Importancia	1

Tabla 6 RUC-05

Requisito	RUC-05
Descripción	En la consulta por enfermedad el usuario podrá seleccionar una enfermedad de una lista
Fuente	Alumno
Fit Criterion	Estar en “Consulta por enfermedad” y pulsar en el desplegable de enfermedades.
Prioridad	Media
Importancia	1

Tabla 7 RUC-06

Requisito	RUC-06
Descripción	En la consulta por enfermedad el usuario podrá seleccionar una fecha de inicio para el intervalo temporal de la consulta
Fuente	Alumno
Fit Criterion	Estar en “Consulta por enfermedad” y pulsar en el calendario interactivo de fecha de inicio.
Prioridad	Media



Importancia	1
--------------------	---

Tabla 8 RUC-07

Requisito	RUC-07
Descripción	En la consulta por enfermedad el usuario podrá seleccionar una fecha de fin para el intervalo temporal de la consulta
Fuente	Alumno
Fit Criterion	Estar en “Consulta por enfermedad” y pulsar en el calendario interactivo de fecha de fin.
Prioridad	Media
Importancia	1

Tabla 9 RUC-08

Requisito	RUC-08
Descripción	El usuario podrá acceder a los resultados de la consulta por enfermedad en forma de gráfico de barras
Fuente	Alumno
Fit Criterion	Pulsar consultar tras haber rellenado todos los campos de la pantalla de consulta por enfermedad
Prioridad	Alta
Importancia	1

Tabla 10 RUC-09

Requisito	RUC-09
Descripción	En la consulta por país el usuario podrá seleccionar un país de una lista
Fuente	Alumno



Fit Criterion	Estar en “Consulta por país” y pulsar en el desplegable de países.
Prioridad	Media
Importancia	1

Tabla 11 RUC-10

Requisito	RUC-10
Descripción	En la consulta por país el usuario podrá establecer una fecha de inicio para el intervalo temporal de la consulta
Fuente	Alumno
Fit Criterion	Estar en “Consulta por país” y pulsar en el calendario interactivo de fecha de inicio.
Prioridad	Media
Importancia	1

Tabla 12 RUC-11

Requisito	RUC-11
Descripción	En la consulta por país el usuario podrá establecer una fecha de fin para el intervalo temporal de la consulta
Fuente	Alumno
Fit Criterion	Estar en “Consulta por país” y pulsar en el calendario interactivo de fecha de fin.
Prioridad	Media
Importancia	1

Tabla 13 RUC-12

Requisito	RUC-12
Descripción	El usuario podrá acceder a los resultados de la consulta por país en forma de gráfico de tarta



Fuente	Alumno
Fit Criterion	Pulsar consultar tras haber rellenado todos los campos de la pantalla de consulta por país
Prioridad	Alta
Importancia	1

Tabla 14 RUC-13

Requisito	RUC-13
Descripción	En la consulta del historial país/enfermedad el usuario podrá seleccionar un país de una lista
Fuente	Alumno
Fit Criterion	Estar en “Consultar historial País/Enfermedad” y pulsar el desplegable de país.
Prioridad	Media
Importancia	1

Tabla 15 RUC-14

Requisito	RUC-14
Descripción	En la consulta del historial país/enfermedad el usuario podrá seleccionar una enfermedad de una lista
Fuente	Alumno
Fit Criterion	Estar en “Consultar historial País/Enfermedad” y pulsar el desplegable de enfermedades.
Prioridad	Media
Importancia	1

Tabla 16 RUC-15

Requisito	RUC-15
------------------	---------------



Descripción	En la consulta del historial por país/enfermedad el usuario podrá establecer una fecha de inicio para el intervalo temporal de la consulta
Fuente	Alumno
Fit Criterion	Estar en “Consultar historial País/Enfermedad” y pulsar en el calendario interactivo de fecha de inicio.
Prioridad	Media
Importancia	1

Tabla 17 RUC-16

Requisito	RUC-16
Descripción	En la consulta del historial por país/enfermedad el usuario podrá establecer una fecha de fin para el intervalo temporal de la consulta
Fuente	Alumno
Fit Criterion	Estar en “Consultar historial País/Enfermedad” y pulsar en el calendario interactivo de fecha de fin.
Prioridad	Media
Importancia	1

Tabla 18 RUC-17

Requisito	RUC-17
Descripción	El usuario podrá acceder a los resultados de la consulta del historial país/enfermedad en forma de gráfica con ejes X e Y.
Fuente	Alumno
Fit Criterion	Pulsar consultar tras haber rellenado todos los campos de la pantalla de consulta del historial país/enfermedad



Prioridad	Alta
Importancia	1

Tabla 19 RUC-18

Requisito	RUC-18
Descripción	El usuario podrá volver a la pantalla anterior a la pantalla de resultados pulsando el botón atrás del smartphone.
Fuente	Decisión conjunta
Fit Criterion	Estar en la pantalla de resultados y pulsar "atrás"
Prioridad	Alta
Importancia	2

Tabla 20 RUC-19

Requisito	RUC-19
Descripción	En cada pantalla de consulta el usuario podrá seleccionar la opción de realizar la consulta sólo con tweets con localización
Fuente	Tutor
Fit Criterion	Estar en una de las pantallas de consulta
Prioridad	Media
Importancia	2

Tabla 21 RUC-20

Requisito	RUC-20
Descripción	El usuario podrá acceder a una sección FAQ



Fuente	Tutor
Fit Criterion	Pulsar sobre el botón de FAQ
Prioridad	Media
Importancia	2

3.6.3 Requisitos de usuario de restricción

Tabla 22 RUR-01

Requisito	RUR-01
Descripción	El usuario no podrá poner como fecha de inicio o de fin la fecha del día actual
Fuente	Alumno
Fit Criterion	Poner fechas en las consultas
Prioridad	Media
Importancia	2

Tabla 23 RUR-02

Requisito	RUR-02
Descripción	El usuario deberá seleccionar dos fechas antes de poder realizar la consulta
Fuente	Alumno
Fit Criterion	Realizar consulta con fechas seleccionadas
Prioridad	Media
Importancia	2

Tabla 24 RUR-03

Requisito	RUC-03
------------------	---------------



Descripción	El sólo tendrá acceso a una actividad a la vez
Fuente	Alumno
Fit Criterion	Ejecutar una actividad
Prioridad	Media
Importancia	2

Tabla 25 RUR-04

Requisito	RUC-04
Descripción	El usuario solo podrá elegir un país o enfermedad en la lista desplegable
Fuente	Alumno
Fit Criterion	Elegir un país o enfermedad en las listas desplegables.
Prioridad	Media
Importancia	2

3.7 Casos de Uso

Los casos de uso vienen dados por los requisitos de usuario. A pesar de que en esta sección se podrían explicar casos de uso de la parte del servidor, como no hay un actor/usuario claro, y el funcionamiento es automático, la explicación del funcionamiento del servidor se dará en secciones posteriores. En esta sección se muestran los casos de uso a nivel de la aplicación de Android obtenidos a partir de sus requisitos de usuario. Se pueden apreciar también diagramas para un entendimiento más sencillo del flujo de los casos de uso.

De la misma manera que en la sección de requisitos de usuario se especifican los campos que componían cada requisito, a continuación se describen los campos que conforman cada caso de uso.

Caso de uso: Nombre autoexplicativo del propio caso de uso

ID: identificador del caso de uso

Nivel de importancia: Con valores que pueden ser Baja, Media y Alta definen la importancia del funcionamiento correcto del caso de uso.

Actor/Actores: usuario/s o tipo/s de usuario/s que participa/n en el caso de uso.

Requisitos de usuario: identificadores de los requisitos de usuario satisfechos en el caso de uso

Descripción: Breve descripción del caso de uso



Trigger: Acción que desencadena la ejecución del caso de uso.

Secuencia normal: Orden de las acciones que se deben realizar para la correcta ejecución del caso de uso.

Secuencia alternativa: Caminos alternativos para la ejecución del caso de uso.

Tabla 26 CU

Caso de uso	
Nivel de importancia	
ID	CU-XX
Actor/Actores	
Requisitos de usuario	
Descripción	
Trigger	
Secuencia normal	
Secuencia alternativa	

3.7.1 Caso de uso principal

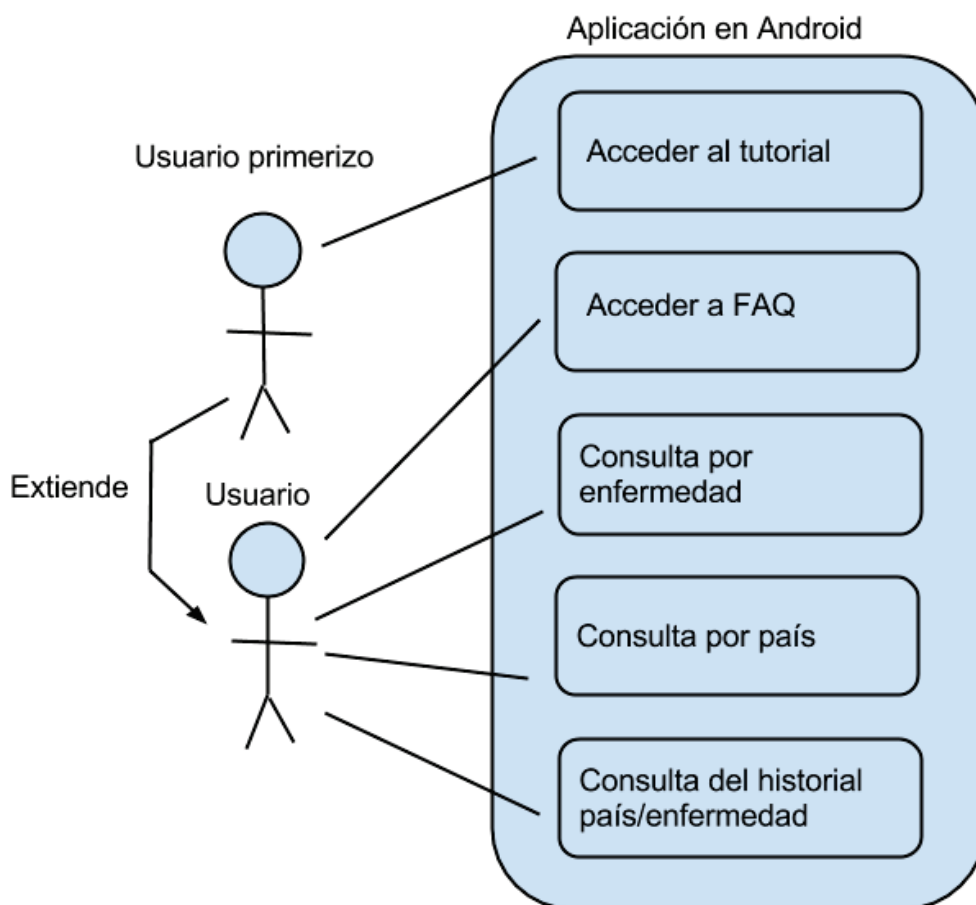


Ilustración 15 Caso de uso principal

Como se puede apreciar en el diagrama hay dos actores que se pueden diferenciar. Por un lado está el usuario primerizo que es el que ejecuta por primera vez la aplicación y es el único tipo de usuario que tiene acceso a al tutorial inicial. Y por otro lado se encuentra el usuario general que es el que puede utilizar todas las funcionalidades de la aplicación. Al no tener la aplicación aspectos más diferenciados en cuanto a comportamiento se refiere no se ha considerado incluir ningún otro tipo de actor. Posteriormente se detallarán los casos 3 últimos casos de uso que aparecen en la figura superior, pero en esta sección se definirán a grandes rasgos los casos de uso que comprenden el caso de uso principal.

Tabla 27 CU-01

Caso de uso	Acceso a tutorial
Nivel de importancia	Media
ID	CU-01



Actor/Actores	Usuario primerizo
Requisitos de usuario	RUC-01
Descripción	Al acceder a la aplicación por primera vez al usuario se le mostrará un breve tutorial
Trigger	Ejecutar la aplicación por primera vez
Secuencia normal	-Ejecutar la aplicación por primera vez -Acceder al tutorial
Secuencia alternativa	

Tabla 28 CU-02

Caso de uso	Acceso a FAQ
Nivel de importancia	Media
ID	CU-02
Actor/Actores	Usuario
Requisitos de usuario	RUC-20
Descripción	Al acceder a la aplicación por primera vez al usuario se le mostrará un breve tutorial
Trigger	Ejecutar la aplicación por primera vez
Secuencia normal	-Ejecutar la aplicación -Encontrarse en el menú principal -Pulsar el botón "FAQ"
Secuencia alternativa	-Encontrarse realizando una consulta -Volver al menú principal -Pulsar el botón "FAQ"

Tabla 29 CU-03

Caso de uso	Consultar por enfermedad
Nivel de importancia	Alta
ID	CU-03
Actor/Actores	Usuario



Requisitos de usuario	RUC-02
Descripción	El usuario podrá acceder a la consulta por enfermedad
Trigger	Pulsar el botón "Consulta por enfermedad"
Secuencia normal	-Ejecutar la aplicación -Encontrarse en el menú principal -Pulsar el botón "Consulta por enfermedad"
Secuencia alternativa	-Encontrarse realizando una consulta -Volver al menú principal -Pulsar el botón "Consulta por enfermedad"

Tabla 30 CU-04

Caso de uso	Consultar por país
Nivel de importancia	Alta
ID	CU-04
Actor/Actores	Usuario
Requisitos de usuario	RUC-03, RUC-09
Descripción	El usuario podrá acceder a la consulta por enfermedad
Trigger	Pulsar el botón "Consulta por país"
Secuencia normal	-Ejecutar la aplicación -Encontrarse en el menú principal -Pulsar el botón "Consulta por país"
Secuencia alternativa	-Encontrarse realizando una consulta -Volver al menú principal -Pulsar el botón "Consulta por país"

Tabla 31 CU-05

Caso de uso	Consultar el historial país/enfermedad
Nivel de importancia	Alta
ID	CU-05



Actor/Actores	Usuario
Requisitos de usuario	RUC-04
Descripción	El usuario podrá acceder a la consulta por enfermedad
Trigger	Pulsar el botón "Consulta historial país/enfermedad"
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Encontrarse en el menú principal-Pulsar el botón "Consulta historial país/enfermedad"
Secuencia alternativa	<ul style="list-style-type: none">-Encontrarse realizando una consulta-Volver al menú principal-Pulsar el botón "Consulta historial país/enfermedad"

3.7.2 Caso de uso de consulta por enfermedad

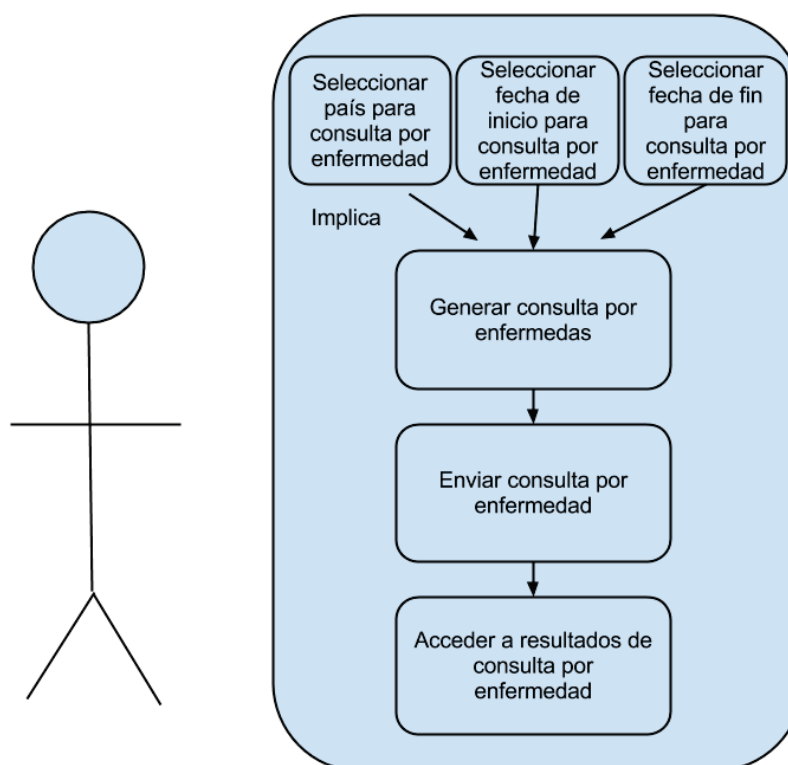


Ilustración 16 Caso de uso consulta por enfermedad

Tal y como se puede apreciar en la imagen superior el caso de uso de consulta por enfermedad conlleva a su vez varios casos de uso que lo conforman. Para generar una consulta por enfermedad hay que elegir una enfermedad, seleccionar una fecha de inicio y seleccionar una fecha de fin. Sólo se ejecuta una vez se han rellenado todos esos campos. A continuación se explican los casos de uso mostrados en la figura anterior a este párrafo.

Tabla 32 CU-06

Caso de uso	Seleccionar enfermedad
Nivel de importancia	Alta
ID	CU-06
Actor/Actores	Usuario
Requisitos de usuario	RUC-02, RUC-05
Descripción	El usuario elige una enfermedad de la lista de enfermedades



Trigger	Pulsar sobre la lista extensible de enfermedades
Secuencia normal	-Ejecutar la aplicación - Acceder a “Consulta por enfermedad” -Pulsar sobre la lista desplegable de enfermedades
Secuencia alternativa	-Encontrarse realizando una consulta -Volver al menú principal -Acceder a “Consulta por enfermedad” -Pulsar sobre la lista desplegable de enfermedades

Tabla 33 CU-07

Caso de uso	Seleccionar fecha de inicio para consulta por enfermedad
Nivel de importancia	Alta
ID	CU-07
Actor/Actores	Usuario
Requisitos de usuario	RUC-06
Descripción	El usuario podrá seleccionar una fecha de inicio para la consulta por enfermedad
Trigger	Pulsar sobre el botón “Seleccionar fecha inicio” en consulta por enfermedad
Secuencia normal	-Ejecutar la aplicación -Acceder a “Consulta por enfermedad” -Pulsar sobre el botón “Seleccionar fecha inicio” -Elegir una fecha -Pulsar sobre el botón “definir”
Secuencia alternativa	-Encontrarse realizando una consulta -Volver al menú principal -Acceder a “Consulta por enfermedad” -Pulsar sobre el botón “Seleccionar fecha inicio” -Elegir una fecha -Pulsar sobre el botón “definir”

Tabla 34 CU-08

Caso de uso	Seleccionar fecha de fin para consulta por enfermedad
Nivel de importancia	Alta
ID	CU-08
Actor/Actores	Usuario
Requisitos de usuario	RUC-07
Descripción	El usuario podrá seleccionar una fecha de fin para la consulta por enfermedad
Trigger	Pulsar sobre el botón “Seleccionar fecha fin” en consulta por enfermedad
Secuencia normal	<ul style="list-style-type: none"> -Ejecutar la aplicación -Acceder a “Consulta por enfermedad” -Pulsar sobre el botón “Seleccionar fecha fin” -Elegir una fecha -Pulsar sobre el botón “definir”
Secuencia alternativa	<ul style="list-style-type: none"> -Encontrarse realizando una consulta -Volver al menú principal -Acceder a “Consulta por enfermedad” -Pulsar sobre el botón “Seleccionar fecha fin” -Elegir una fecha -Pulsar sobre el botón “definir”

Tabla 35 CU-09

Caso de uso	Generar consulta por enfermedad
Nivel de importancia	Alta
ID	CU-09
Actor/Actores	Usuario
Requisitos de usuario	RUC-08
Descripción	El usuario rellena todos los campos pertinentes y genera una consulta por enfermedad
Trigger	Haber rellenado todos los campos
Secuencia normal	-Ejecutar la aplicación



	<ul style="list-style-type: none">-Acceder a "Consulta por enfermedad"-Seleccionar una enfermedad-Seleccionar una fecha de inicio-Seleccionar una fecha de fin
Secuencia alternativa	

Tabla 36 CU-10

Caso de uso	Enviar consulta por enfermedad
Nivel de importancia	Alta
ID	CU-10
Actor/Actores	Usuario
Requisitos de usuario	RUC-08, RUC-02
Descripción	Una vez generada una consulta el usuario la envía
Trigger	Haber rellenado todos los campos y pulsar sobre el botón "Consultar"
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta por enfermedad"-Seleccionar una enfermedad-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"
Secuencia alternativa	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta por enfermedad"-Seleccionar una enfermedad-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"-El número de días de diferencia entre la fecha de inicio y la fecha de fin es superior a 120-Se muestra mensaje de error



Tabla 37 CU-11

Caso de uso	Acceder a los resultados de una consulta por enfermedad
Nivel de importancia	Alta
ID	CU-11
Actor/Actores	Usuario
Requisitos de usuario	RUC-08
Descripción	Tras enviar la consulta y esperar la respuesta el usuario accede automáticamente a los resultados
Trigger	Pulsar botón consultar
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta por enfermedad"-Seleccionar una enfermedad-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"-Aparición por pantalla de los resultados
Secuencia alternativa	

3.7.3 Caso de uso de consulta por país

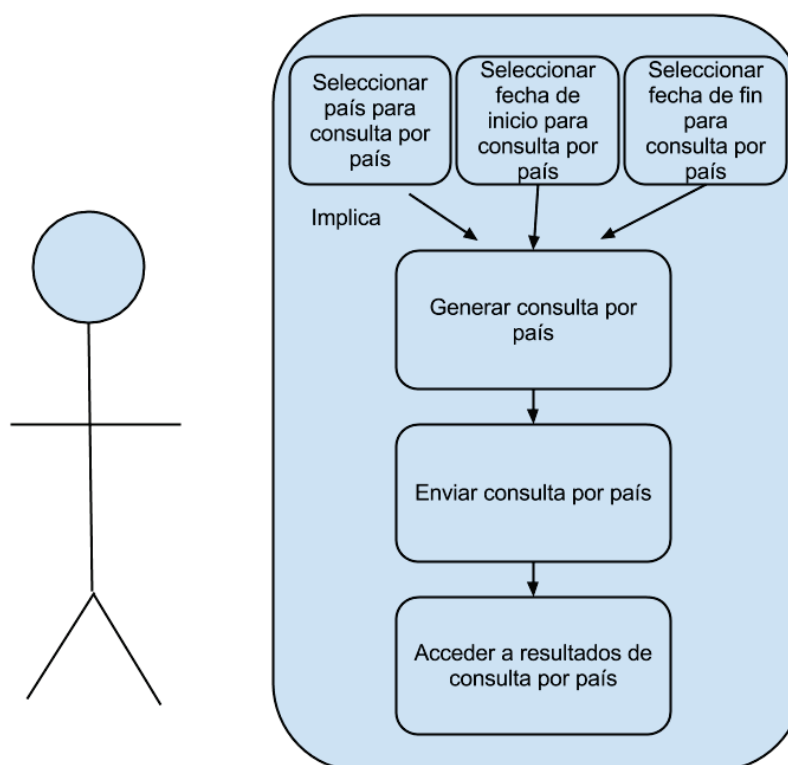


Ilustración 17 Caso de uso consulta por país

Este caso de uso es igual al caso de uso anterior pero con la diferencia que en vez de realizar la consulta por enfermedad se realiza por país. El actor sigue siendo el usuario. Los casos de uso que componen este módulo son los siguientes.

Tabla 38 CU-12

Caso de uso	Seleccionar país
Nivel de importancia	Alta
ID	CU-12
Actor/Actores	Usuario
Requisitos de usuario	RUC-09
Descripción	El usuario elige un país de la lista de países
Trigger	Pulsar sobre la lista extensible de países y seleccionar un país



Secuencia normal	-Ejecutar la aplicación - Acceder a “Consulta por país” -Pulsar sobre la lista desplegable de país -Seleccionar un país
Secuencia alternativa	-Encontrarse realizando una consulta -Volver al menú principal -Acceder a “Consulta por país” -Pulsar sobre la lista desplegable de países -Seleccionar un país

Tabla 39 CU-13

Caso de uso	Seleccionar fecha de inicio para consulta por país
Nivel de importancia	Alta
ID	CU-13
Actor/Actores	Usuario
Requisitos de usuario	RUC-10
Descripción	El usuario podrá seleccionar una fecha de inicio para la consulta por país
Trigger	Pulsar sobre el botón “Seleccionar fecha inicio” en consulta por país
Secuencia normal	-Ejecutar la aplicación -Acceder a “Consulta por país” -Pulsar sobre el botón “Seleccionar fecha inicio” -Elegir una fecha -Pulsar sobre el botón “definir”
Secuencia alternativa	-Encontrarse realizando una consulta -Volver al menú principal -Acceder a “Consulta por país” -Pulsar sobre el botón “Seleccionar fecha inicio” -Elegir una fecha -Pulsar sobre el botón “definir”

Tabla 40 CU-14

Caso de uso	Seleccionar fecha de fin para consulta por país
Nivel de importancia	Alta



ID	CU-14
Actor/Actores	Usuario
Requisitos de usuario	RUC-11
Descripción	El usuario podrá seleccionar una fecha de fin para la consulta por país
Trigger	Pulsar sobre el botón “Seleccionar fecha fin” en consulta por país
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a “Consulta por país”-Pulsar sobre el botón “Seleccionar fecha fin”-Elegir una fecha-Pulsar sobre el botón “definir”
Secuencia alternativa	<ul style="list-style-type: none">-Encontrarse realizando una consulta-Volver al menú principal-Acceder a “Consulta por país”-Pulsar sobre el botón “Seleccionar fecha fin”-Elegir una fecha-Pulsar sobre el botón “definir”

Tabla 41 CU-15

Caso de uso	Generar consulta por país
Nivel de importancia	Alta
ID	CU-15
Actor/Actores	Usuario
Requisitos de usuario	RUC-12
Descripción	El usuario rellena todos los campos pertinentes y genera una consulta por país
Trigger	Haber rellenado todos los campos
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a “Consulta por país”-Seleccionar una país-Seleccionar una fecha de inicio-Seleccionar una fecha de fin
Secuencia alternativa	



--	--

Tabla 42 CU-16

Caso de uso	Enviar consulta por país
Nivel de importancia	Alta
ID	CU-16
Actor/Actores	Usuario
Requisitos de usuario	RUC-12
Descripción	Una vez generada una consulta el usuario la envía
Trigger	Haber rellenado todos los campos y pulsar sobre el botón "Consultar"
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta por país"-Seleccionar una país-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"
Secuencia alternativa	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta por país"-Seleccionar una país-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"-El número de días de diferencia entre la fecha de inicio y la fecha de fin es superior a 120-Se muestra mensaje de error

Tabla 43 CU-17

Caso de uso	Acceder a los resultados de una consulta por país
Nivel de importancia	Alta
ID	CU-17
Actor/Actores	Usuario



Requisitos de usuario	RUC-12
Descripción	Tras enviar la consulta y esperar la respuesta el usuario accede automáticamente a los resultados
Trigger	Pulsar botón consultar
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta por país"-Seleccionar una enfermedad-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"-Aparición por pantalla de los resultados
Secuencia alternativa	

3.7.3 Caso de uso de consulta por el historial país/enfermedad

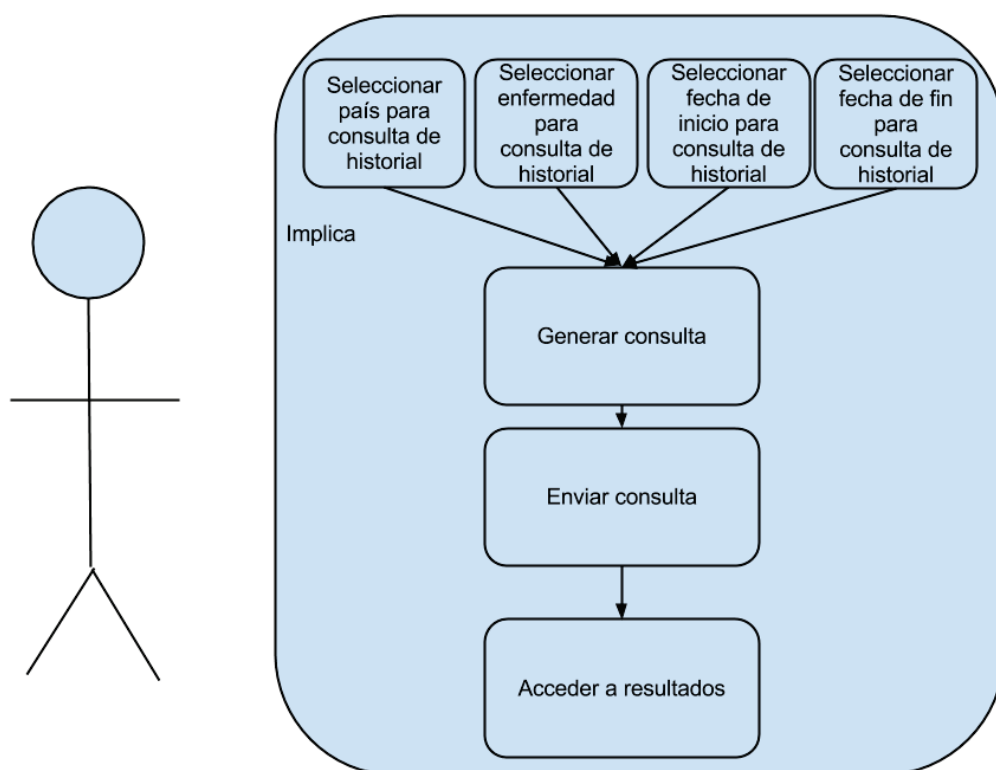


Ilustración 18 Caso de uso consulta de historial

Este caso de uso en concreto mantiene el mismo actor que los anteriores pero incluye las diferencias que tenían los anteriores. Los casos de uso que conforman este caso de uso general son:

Tabla 44 CU-18

Caso de uso	Seleccionar país para consulta de historial país/enfermedad
Nivel de importancia	Alta
ID	CU-18
Actor/Actores	Usuario
Requisitos de usuario	RUC-13
Descripción	El usuario elige un país de la lista de países



Trigger	Pulsar sobre la lista extensible de países y seleccionar un país
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación- Acceder a “Consulta de historial país/enfermedad”-Pulsar sobre la lista desplegable de país-Seleccionar un país
Secuencia alternativa	<ul style="list-style-type: none">-Encontrarse realizando una consulta-Volver al menú principal-Acceder a “Consulta de historial país/enfermedad”-Pulsar sobre la lista desplegable de países-Seleccionar un país

Tabla 45 CU-19

Caso de uso	Seleccionar enfermedad para consulta de historial país/enfermedad
Nivel de importancia	Alta
ID	CU-19
Actor/Actores	Usuario
Requisitos de usuario	RUC-14
Descripción	El usuario elige una enfermedad de la lista de enfermedades
Trigger	Pulsar sobre la lista extensible de enfermedades
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación- Acceder a “Consulta de historial país/enfermedad”-Pulsar sobre la lista desplegable de enfermedades
Secuencia alternativa	<ul style="list-style-type: none">-Encontrarse realizando una consulta-Volver al menú principal-Acceder a “Consulta de historial país/enfermedad”-Pulsar sobre la lista desplegable de enfermedades

Tabla 46 CU-20

Caso de uso	Seleccionar fecha de inicio para consulta de historial país/enfermedad
Nivel de importancia	Alta
ID	CU-20
Actor/Actores	Usuario
Requisitos de usuario	RUC-15
Descripción	El usuario podrá seleccionar una fecha de inicio para la consulta de historial país/enfermedad
Trigger	Pulsar sobre el botón “Seleccionar fecha inicio” en consulta de historial país/enfermedad
Secuencia normal	<ul style="list-style-type: none"> -Ejecutar la aplicación -Acceder a “Consulta de historial país/enfermedad” -Pulsar sobre el botón “Seleccionar fecha inicio” -Elegir una fecha -Pulsar sobre el botón “definir”
Secuencia alternativa	

Tabla 47 CU-21

Caso de uso	Seleccionar fecha de fin para consulta de historial país/enfermedad
Nivel de importancia	Alta
ID	CU-21
Actor/Actores	Usuario
Requisitos de usuario	RUC-16
Descripción	El usuario podrá seleccionar una fecha de fin para la consulta de historial país/enfermedad
Trigger	Pulsar sobre el botón “Seleccionar fecha fin” en consulta de historial país/enfermedad
Secuencia normal	-Ejecutar la aplicación



	-Acceder a “Consulta de historial país/enfermedad” -Pulsar sobre el botón “Seleccionar fecha fin” -Elegir una fecha -Pulsar sobre el botón “definir”
Secuencia alternativa	

Tabla 48 CU-22

Caso de uso	Generar consulta de historial país/enfermedad
Nivel de importancia	Alta
ID	CU-22
Actor/Actores	Usuario
Requisitos de usuario	RUC-17
Descripción	El usuario rellena todos los campos pertinentes y genera una consulta de historial país/enfermedad
Trigger	Haber rellenado todos los campos
Secuencia normal	-Ejecutar la aplicación -Acceder a “Consulta de historial país/enfermedad” -Seleccionar una país -Seleccionar enfermedad -Seleccionar una fecha de inicio -Seleccionar una fecha de fin
Secuencia alternativa	

Tabla 49 CU-23

Caso de uso	Enviar consulta de historial país/enfermedad
Nivel de importancia	Alta
ID	CU-23
Actor/Actores	Usuario
Requisitos de usuario	RUC-17



Descripción	Una vez generada una consulta el usuario la envía
Trigger	Haber rellenado todos los campos y pulsar sobre el botón "Consultar"
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta de historial país/enfermedad"-Seleccionar una país-Seleccionar una enfermedad-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"
Secuencia alternativa	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta de historial país/enfermedad"-Seleccionar una país-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"-El número de días de diferencia entre la fecha de inicio y la fecha de fin es superior a 120-Se muestra mensaje de error

Tabla 50 CU-24

Caso de uso	Acceder a los resultados de una consulta de historial país/enfermedad
Nivel de importancia	Alta
ID	CU-24
Actor/Actores	Usuario
Requisitos de usuario	RUC-17
Descripción	Tras enviar la consulta y esperar la respuesta el usuario accede automáticamente a los resultados
Trigger	Pulsar botón consultar
Secuencia normal	<ul style="list-style-type: none">-Ejecutar la aplicación-Acceder a "Consulta de historial país/enfermedad"



	<ul style="list-style-type: none">-Seleccionar un país-Seleccionar una enfermedad-Seleccionar una fecha de inicio-Seleccionar una fecha de fin-Pulsar botón "Consultar"-Aparición por pantalla de los resultados
Secuencia alternativa	

3.8 Requisitos de Software

Los requisitos de software son aquellos requisitos que dictaminan qué debe hacer el sistema. Se definen en función de los casos de uso y los requisitos de usuario. En esta ocasión al ser un proyecto compuesto por un servidor y una aplicación se detallarán por separado los requisitos de software de cada parte. Y, a pesar de no haber definido requisitos de usuario ni casos de uso para la parte del servidor, puesto que no existen actores, aquí se definirán los requisitos de software de esa parte para que se pueda comprender mejor el funcionamiento del proyecto entero. Los requisitos se dividirán en funcionales y no funcionales. Los primeros son los que especifican el qué y los segundos el cómo.

Como en anteriores secciones los requisitos se mostrarán en forma de tablas que contienen una serie de atributos descritos a continuación:

- **Requisito:** es el identificador único del requisito de software se nombrarán tanto con siglas (RSF o RSNF) como con números
- **Descripción:** una breve descripción del requisito.
- **Nivel de importancia:** Alta, Media o Baja determinan la importancia de la implementación de este requisito.
- **Requisitos de Usuario:** Campo en el que se especifican los requisitos de usuario que sientan la base para este requisito de software. En algunos casos no habrá requisitos que sean base puesto que se trata de tareas que hacen que funcione todo el sistema.

Las tablas serán similares a la que se expone en la siguiente figura:

Tabla 51 RSF/RSNF

Requisito	RSF/RSNF-XX
Descripción	
Nivel de importancia	



Requisitos de Usuario	
------------------------------	--

3.8.1 Requisitos funcionales (aplicación Android)

Tabla 52 RSF-01

Requisito	RSF-01
Descripción	La aplicación será instalable en sistema operativo Android
Nivel de importancia	Alta
Requisitos de Usuario	-

Tabla 53 RSF-02

Requisito	RSF-02
Descripción	La aplicación mostrará un tutorial en su primera ejecución
Nivel de importancia	Media
Requisitos de Usuario	RUC-01

Tabla 54 RSF-03

Requisito	RSF-03
Descripción	La aplicación permitirá al usuario acceder al FAQ
Nivel de importancia	Media
Requisitos de Usuario	RUC-20

Tabla 55 RSF-04

Requisito	RSF-04
------------------	---------------



Descripción	La aplicación permitirá al usuario volver desde cualquier pantalla a la anterior
Nivel de importancia	Alta
Requisitos de Usuario	RUC-18

Tabla 56 RSF-05

Requisito	RSF-05
Descripción	La aplicación permitirá al usuario hacer consultas por enfermedad
Nivel de importancia	Alta
Requisitos de Usuario	RUC-02

Tabla 57 RSF-06

Requisito	RSF-06
Descripción	La aplicación permitirá al usuario hacer consultas por país
Nivel de importancia	Alta
Requisitos de Usuario	RUC-03

Tabla 58 RSF-07

Requisito	RSF-07
Descripción	La aplicación permitirá al usuario hacer consultas del historial país/enfermedad
Nivel de importancia	Alta
Requisitos de Usuario	RUC-04

Tabla 59 RSF-08

Requisito	RSF-08
------------------	---------------



Descripción	La aplicación permitirá al usuario elegir un país de una lista para las consultas por país
Nivel de importancia	Media
Requisitos de Usuario	RUC-09

Tabla 60 RSF-09

Requisito	RSF-09
Descripción	La aplicación permitirá al usuario elegir una enfermedad de una lista para las consultas por enfermedad
Nivel de importancia	Media
Requisitos de Usuario	RUC-05

Tabla 61 RSF-10

Requisito	RSF-10
Descripción	La aplicación permitirá al usuario escoger un país y una enfermedad para las consultas de historial país/enfermedad
Nivel de importancia	Media
Requisitos de Usuario	RUC-13, RUC-14

Tabla 62 RSF-11

Requisito	RSF-11
Descripción	La aplicación permitirá al usuario configurar una fecha de inicio para la consulta por enfermedad
Nivel de importancia	Media
Requisitos de Usuario	RUC-06



Tabla 63 RSF-12

Requisito	RSF-12
Descripción	La aplicación permitirá al usuario configurar una fecha de inicio para la consulta por país
Nivel de importancia	Media
Requisitos de Usuario	RUC-10

Tabla 64 RSF-13

Requisito	RSF-13
Descripción	La aplicación permitirá al usuario configurar una fecha de inicio para la consulta del historial país/enfermedad
Nivel de importancia	Media
Requisitos de Usuario	RUC-15

Tabla 65 RSF-14

Requisito	RSF-14
Descripción	La aplicación permitirá al usuario configurar una fecha de fin para las consultas por país
Nivel de importancia	Media
Requisitos de Usuario	RUC-10

Tabla 66 RSF-15

Requisito	RSF-15
Descripción	La aplicación permitirá al usuario configurar una fecha de fin para las consultas por enfermedad
Nivel de importancia	Media
Requisitos de Usuario	RUC-07



Tabla 67 RSF-17

Requisito	RSF-16
Descripción	La aplicación permitirá al usuario configurar una fecha de fin para las consultas del historial país enfermedad
Nivel de importancia	Media
Requisitos de Usuario	RUC-16

Tabla 68 RSF-17

Requisito	RSF-17
Descripción	La aplicación permitirá al usuario elegir entre tweets con localización o sin localización para cualquier consulta
Nivel de importancia	Alta
Requisitos de Usuario	RUC-19

Tabla 69 RSF-18

Requisito	RSF-18
Descripción	La aplicación mostrará al usuario acceder a los resultados de la consulta por enfermedad
Nivel de importancia	Alta
Requisitos de Usuario	RUC-08

Tabla 70 RSF-19

Requisito	RSF-19
Descripción	La aplicación mostrará al usuario acceder a los resultados de consulta por país
Nivel de importancia	Alta
Requisitos de Usuario	RUC-12



Tabla 71 RSF-20

Requisito	RSF-20
Descripción	La aplicación mostrará al usuario acceder a los resultados de consulta del historial país/enfermedad
Nivel de importancia	Alta
Requisitos de Usuario	RUC-17

3.8.2 Requisitos funcionales (Servidor)

A pesar de no existir un usuario claramente definido para el subsistema del servidor, este debe cumplir una serie de requisitos para que todo el sistema funcione correctamente. Estos requisitos se exponen a continuación en forma de las tablas creadas anteriormente para los requisitos de software con la variación de que no existe el campo requisitos de usuario puesto que no satisfacen ningún requisito de usuario directamente si no que permiten que algunas partes de la aplicación los satisfagan, por lo tanto ese campo se transforma en justificación en el sistema donde se explicará brevemente el porqué de ese requisito. Los requisitos de software funcionales del servidor que se han definido para este proyecto son:

Tabla 72 RSF-21

Requisito	RSF-21
Descripción	El sistema ejecutará un programa que recopilará información de Twitter
Nivel de importancia	Alta
Justificación en el sistema	El programa es necesario para poder tener una información que servir a la aplicación

Tabla 73 RSF-22

Requisito	RSF-22
Descripción	El sistema utilizará expresiones relacionadas con enfermedades para realizar las búsquedas
Nivel de importancia	Media



Justificación en el sistema	La búsquedas necesitan una serie de criterios para ser realizadas
------------------------------------	---

Tabla 74 RSF-23

Requisito	RSF-23
Descripción	El sistema contemplará un porcentaje de resultados inválido
Nivel de importancia	Media
Justificación en el sistema	Dado que Twitter es una red social algunas de los resultados encontrados pueden ser “falsos positivos” por lo que hay que ponderar la veracidad de estos resultados.

Tabla 75 RSF-24

Requisito	RSF-24
Descripción	El programa que recopila información trata algunos resultados de las búsquedas realizadas antes de enviarlos a la base de datos
Nivel de importancia	Media
Justificación en el sistema	Algunos tweets pueden no contener información de localización por lo que hay que seguir algún criterio para adjudicarlos a determinados países

Tabla 76 RSF-25

Requisito	RSF-25
Descripción	El sistema almacenará la información recopilada en una base de datos.
Nivel de importancia	Alta
Justificación en el sistema	Una base de datos es por el momento el medio más efectivo para almacenar información que posteriormente va a ser accedida



Tabla 77 RSF-26

Requisito	RSF-26
Descripción	El sistema tendrá dos tablas en la base de datos: una para tweets con localización y otra para tweets sin localización
Nivel de importancia	Alta
Justificación en el sistema	Para hacer más rápidas y sencillas las consultas se implementan dos tablas que no están interrelacionadas

Tabla 78 RSF-27

Requisito	RSF-27
Descripción	El sistema permitirá al programa que recopila la información de Twitter acceder a la base de datos para almacenar la información recopilada.
Nivel de importancia	Alta
Justificación en el sistema	Para que la información pueda ser almacenada debe existir la posibilidad de conexión entre la base de datos y el buscador de información

Tabla 79 RSF-28

Requisito	RSF-28
Descripción	El sistema implementará una interfaz para conexiones externas a la base de datos
Nivel de importancia	Alta
Justificación en el sistema	El sistema debe contar una serie de servicios web para conexiones externas.

Tabla 80 RSF-29

Requisito	RSF-29
Descripción	El sistema proporcionará siempre una respuesta ante cualquier petición



Nivel de importancia	Alta
Justificación en el sistema	Tanto en el caso de existir como en el de no existir la información requerida el sistema debe devolver una respuesta para informar de la situación de la consulta.

Tabla 81 RSF-30

Requisito	RSF-30
Descripción	Las respuestas proporcionadas tendrán un límite de tiempo para proporcionarse
Nivel de importancia	Alta
Justificación en el sistema	Las respuestas no pueden tardar más de un determinado tiempo puesto que si no el uso de la aplicación se haría aburrido por lo tanto debe contemplar un tiempo máximo de respuesta.

Tabla 82 RSF-31

Requisito	RSF-31
Descripción	El sistema implementará un servidor para gestionar las peticiones externas
Nivel de importancia	Alta
Justificación en el sistema	Un sistema de servidor es totalmente necesario para poder interconectar todos los elementos entre sí y además ofrecer un servicio externo.

Tabla 83 RSF-32

Requisito	RSF-32
Descripción	El sistema permitirá a la aplicación móvil acceder a la información recopilada
Nivel de importancia	Alta
Justificación en el sistema	Para que el sistema de servidor tenga sentido debe proporcionar acceso a la información a la aplicación móvil desarrollada



3.8.3 Requisitos no funcionales (aplicación Android)

Tabla 84 RSNF-01

Requisito	RSNF-01
Descripción	La aplicación se ejecutará al pulsar sobre su icono en el menú de aplicaciones
Nivel de importancia	Alta

Tabla 85 RSNF-02

Requisito	RSNF-02
Descripción	La aplicación deberá tener acceso a internet para ejecutar las consultas
Nivel de importancia	Alta

Tabla 86 RSNF-03

Requisito	RSNF-03
Descripción	La interfaz de la aplicación tendrá un aspecto agradable y será sencilla de usar
Nivel de importancia	Alta

Tabla 87 RSNF-04

Requisito	RSNF-04
Descripción	La aplicación será instalable en el sistema operativo Android
Nivel de importancia	Alta

Tabla 88 RSNF-05

Requisito	RSNF-05
------------------	----------------



Descripción	La aplicación mostrará los resultados de consulta por enfermedad en un gráfico de barras
Nivel de importancia	Media

Tabla 89 RSNF-06

Requisito	RSNF-06
Descripción	La aplicación mostrará los resultados de la consulta por país en un gráfico de tarta
Nivel de importancia	Media

Tabla 90 RSNF-07

Requisito	RSNF-07
Descripción	La aplicación mostrará los resultados de la consulta del historial país/enfermedad en una gráfica x y
Nivel de importancia	Media

Tabla 91 RSNF-08

Requisito	RSNF-08
Descripción	La aplicación ofrecerá en forma de checkbox la opción de usar solo tweets con localización
Nivel de importancia	Media

Tabla 92 RSNF-09

Requisito	RSNF-09
Descripción	La aplicación proporcionará un calendario interactivo para configurar la fecha de inicio de las consultas
Nivel de importancia	Media



Tabla 93 RSNF-10

Requisito	RSNF-10
Descripción	La aplicación mostrará un calendario interactivo para configurar la fecha de fin de las consultas
Nivel de importancia	Media

Tabla 94 RSNF-11

Requisito	RSNF-11
Descripción	La lista de países en las consultas será una lista despegable
Nivel de importancia	Media

Tabla 95 RSNF-12

Requisito	RSNF-12
Descripción	La lista de enfermedades en las consultas será una lista desplegable
Nivel de importancia	Media

Tabla 96 RSNF-13

Requisito	RSNF-13
Descripción	Las consultas se resolverán en un tiempo menor a 15 segundos
Nivel de importancia	Alta
Requisitos de Usuario	

Tabla 97 RSNF-14

Requisito	RSNF-14
Descripción	La aplicación mostrará un mensaje durante la ejecución de la consulta
Nivel de importancia	Media



Tabla 98 RSNF-15

Requisito	RSNF-15
Descripción	La aplicación se mostrará en castellano
Nivel de importancia	Media

Tabla 99 RSNF-16

Requisito	RSNF-16
Descripción	La aplicación mostrará un mensaje si algún campo no se ha rellenado
Nivel de importancia	Alta

Tabla 100 RSNF-17

Requisito	RSNF-17
Descripción	La aplicación no permitirá realizar consultas si un campo no se ha rellenado
Nivel de importancia	Alta



3.8.4 Requisitos no funcionales (Servidor)

Tabla 101 RSNF-18

Requisito	RSNF-18
Descripción	El servidor será Apache
Nivel de importancia	Alta

Tabla 102 RSNF-19

Requisito	RSNF-19
Descripción	La base de datos funcionará bajo MySQL
Nivel de importancia	Alta

Tabla 103 RSNF-20

Requisito	RSNF-20
Descripción	El programa de búsqueda de tweets se desarrollará en Java
Nivel de importancia	Alta

Tabla 104 RSNF-21

Requisito	RSNF-21
Descripción	El programa en Java utilizará la librería twitter4j para el acceso a Twitter.
Nivel de importancia	Alta

Tabla 105 RSNF-22

Requisito	RSNF-22
-----------	---------



Descripción	Los servicios para manejo de la base de datos se realizará a través de archivos PHP
Nivel de importancia	Alta

Tabla 106 RSNF-23

Requisito	RSNF-23
Descripción	El programa buscador se ejecutará varias veces a lo largo del día
Nivel de importancia	Alta

Tabla 107 RSNF-24

Requisito	RSNF-24
Descripción	Sólo habrá una base de datos
Nivel de importancia	Alta

Tabla 108 RSNF-25

Requisito	RSNF-25
Descripción	El servicio funcionará en localhost
Nivel de importancia	Alta

4 Diseño

El diseño del sistema se divide en dos partes. Por un lado la parte de la aplicación móvil y por otro la parte del servidor. En esta sección se pretende explicar la arquitectura del sistema y la arquitectura de cada una de las partes de manera detallada.

4.1 Esquema general del sistema

En esta sección se explicará que esquema sigue el sistema y se describirán las principales funcionalidades del mismo. La parte de código y funcionamiento detallado se expondrá en la siguiente sección (implementación). Esta parte pretende explicar con claridad el flujo de los datos dentro del sistema así como su funcionamiento general.

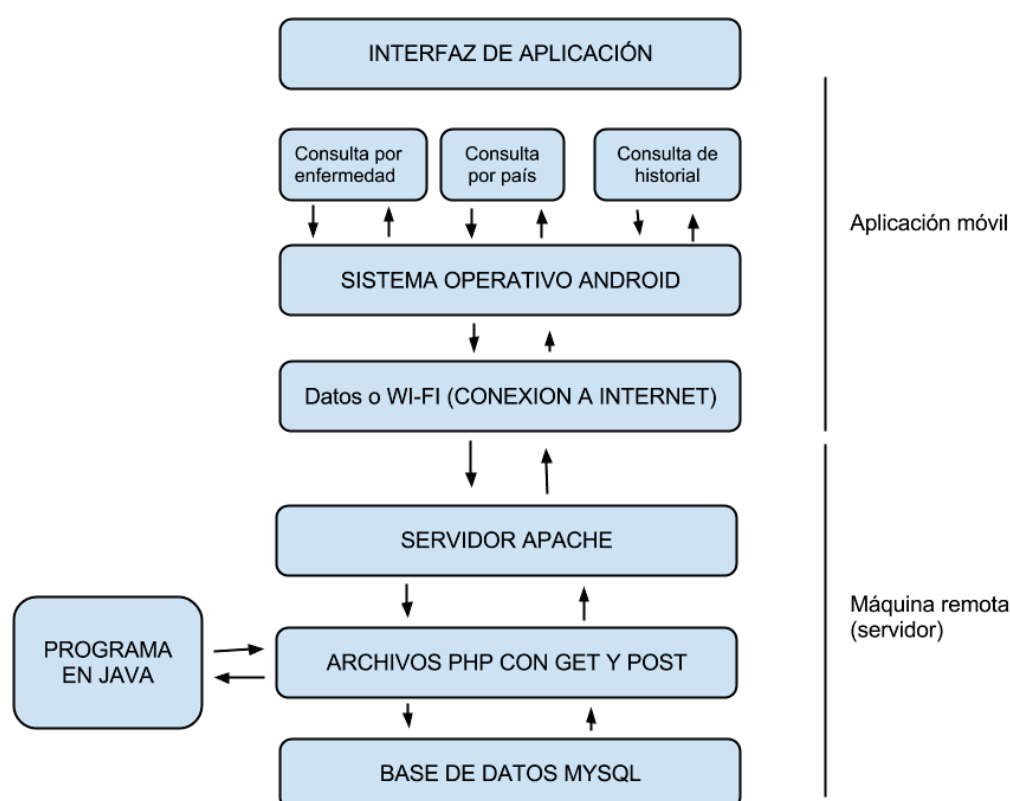


Ilustración 19 Esquema general del sistema

Como se puede observar en la figura el sistema se compone de dos partes principales. Una parte que se ejecuta en forma de aplicación en Android compuesta por varios módulos que son la interfaz, las distintas consultas, el sistema operativo y la conexión a internet. La otra parte es un servidor remoto alojado en mi ordenador personal que hace las veces de servidor compuesta por un servidor apache, ficheros php que sirven como web services con varias



consultas para la base de datos, una base de datos y un programa en java que realiza las consultas a Twitter.

4.2 Aplicación móvil

La aplicación móvil es la parte encargada de conectar al usuario principal con el sistema. Desde aquí se puede acceder a las distintas consultas y posteriores resultados. Es la única parte que el usuario podrá ver por lo que sus puntos fuertes deben ser el diseño y la usabilidad.

El lenguaje de programación de la aplicación es Java debido a que el sistema operativo sobre el que se ejecutará es Android. Java es un lenguaje de programación de alto nivel orientado a objetos. Y, en Android, el código se compila en un ejecutable Dalvik que es una máquina virtual especializada para Android y enfocada a dispositivos móviles.

Para el desarrollo de la aplicación y como se ha mencionado anteriormente se utiliza el entorno de desarrollo eclipse y su plugin ADT.

Al desarrollar una aplicación en Android hay que tener en cuenta 4 elementos principales:

- Las actividades: en Android las actividades son las acciones que se pueden realizar en la aplicación. Se pueden comunicar entre ellas y por lo general cada actividad es representada por una o más pantallas.
- Base de datos: Android proporciona la posibilidad de crear y acceder a una base de datos en el dispositivo móvil. Esta base de datos se suele utilizar para almacenar aquellos datos que requieren ser accedidos muy rápidamente y cuyo volumen no va a ser demasiado grande como para llenar el dispositivo.
- Comunicaciones: cualquier aplicación de Android que se conecta al exterior debe habilitar y configurar una serie de mecanismos para poder llevar a cabo estas comunicaciones.
- Pantallas: de manera casi paralela a las actividades se desarrollan las pantallas. Se desarrollan en código xml aunque Android dispone de una interfaz gráfica para su diseño. Se puede configurar desde el color y la imagen del fondo hasta la transparencia de los botones.

Una vez explicadas brevemente las características generales de una aplicación Android de cara a su desarrollo se procede a particularizar cada punto para el presente proyecto.

En este proyecto se sigue un diagrama de actividades similar al siguiente:

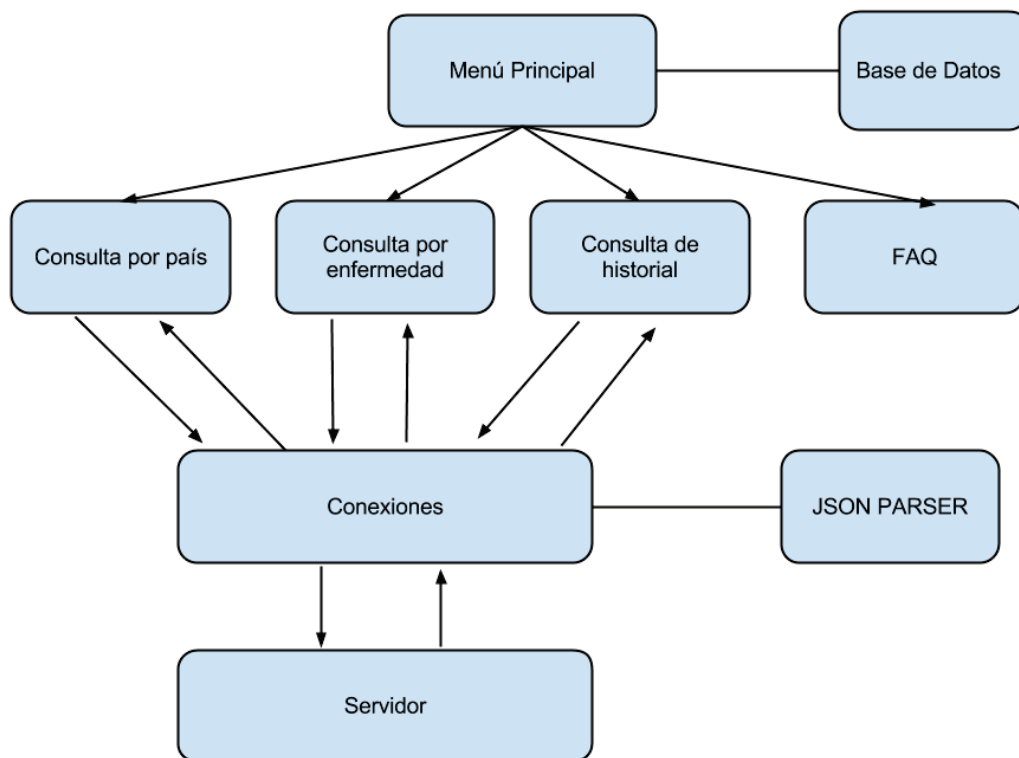


Ilustración 20 Diagrama de actividades

4.2.1 Actividades.

Como se puede observar en el diagrama anterior existe una actividad principal también llamada main activity. Esta actividad es la que se ejecuta al abrir la aplicación. En nuestro caso esa actividad es el menú principal. Del menú principal se puede acceder a 4 actividades distintas. FAQ, consulta por enfermedad, consultas básicas y consulta del historial país/enfermedad. De las 3 últimas se accede a las pantallas de resultados pero que no son actividades propias si no imágenes generadas por la librería encargada de pintar los datos. A cada actividad se accede tras pulsar un botón que indica a donde se va a pasar. Es un diseño muy sencillo dado que la aplicación móvil es sólo una interfaz con la cual el usuario tiene acceso a la representación de unos datos que ya están almacenados.

A continuación se detalla el funcionamiento general de cada actividad.

Menú principal

Es la actividad principal. La que se ejecuta cuando se inicializa la aplicación desde el menú de aplicaciones. Esta actividad contiene 4 conexiones a distintas actividades. Su función sólo es la



de servir de punto de referencia para el usuario y dar acceso al resto de actividades. Pero hay que mencionar que tiene un mecanismo interno para cuando es la primera vez que se accede a la aplicación. Si es la primera vez que se accede a la aplicación esta actividad redirecciona automáticamente al usuario a la sección FAQ que sirve como tutorial para aprender a manejar el resto de funcionalidades. Además varía el valor de la base de datos para hacer que la próxima vez que se ejecute la aplicación no aparezca el tutorial.

FAQ

Esta actividad contiene un texto que explica en detalle cómo funciona la aplicación. Es accedida de dos formas. Por un lado se puede acceder a ella desde el menú principal pulsando sobre un texto resaltado con las letras FAQ y por otro lado se accede directamente cuando se entra por primera vez en la aplicación. Su función es la de resolver las posibles dudas de uso y concepto que pueda tener el usuario al usar la aplicación.

Consultas básicas.

Esta actividad es más compleja que las anteriores ya que se conecta al servidor de la aplicación. Contiene varios campos para configurar la búsqueda a realizar. En esta actividad se realizan dos actividades distintas por un lado se tiene acceso a las consultas por enfermedad y por otro lado se tiene acceso a las consultas por país. Esto se ha hecho así para optimizar en actividades dado que necesitan prácticamente los mismos campos.

Todas las consultas siguen un esquema similar a este:

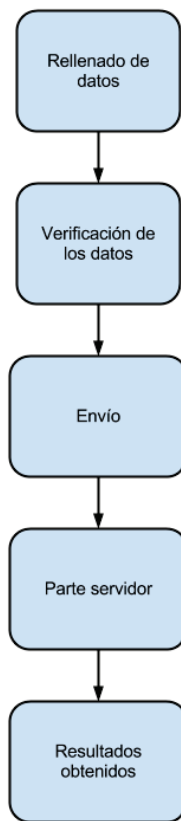


Ilustración 21 Esquema de consultas

Consulta por enfermedad

Implementa una lista de enfermedades que al ser pulsada por el usuario se despliega y le permite elegir una de ellas. Las enfermedades dispuestas en la lista son Alergia, Gastroenteritis, Anginas, Constipado y Gripe. Se han elegido estas enfermedades para desarrollar la aplicación porque son las enfermedades más típicas en cualquier población “occidental” y sobre las que más se habla en Twitter. Una vez elegida la enfermedad se pueden elegir las fechas de inicio y fin de la consulta. Estas elecciones se realizan mediante la implementación de un calendario interactivo que permite al usuario elegir una fecha de una manera clara y precisa. En penúltimo lugar se encuentra un checkbox que permite al usuario utilizar sólo tweets con localización para realizar la consulta o utilizarlos todos. Y por último se encuentra el botón consultar que sirve para llevar a cabo la consulta. Este botón está asociado a varios eventos tales como comprobar que la fecha de fin no es superior a la fecha actual o comprobar que ambos campos de fecha están rellenos. Una vez se pulsa el botón consultar y se han comprobado los campos de la consulta a realizar es el momento de pedir los datos al servidor para conformar la gráfica. Los pasos que se siguen son:

- Se envía la fecha 1 más la enfermedad consultada



- El envío se realiza mediante JSON y la función GET
- Se llama al fichero PHP adecuado y se le pasan como parámetros la fecha y la enfermedad
- Se realiza la consulta en la base de datos que devuelve una lista con países y su acumulado para la fecha inicial
- Al llegar a la aplicación los datos se extraen de un objeto JSONArray y se almacenan en un array
- Se repite el proceso para la segunda fecha
- Se restan los valores de enfermos almacenados en cada array según el país
- El resultado se pasa a la función que pinta la gráfica de barras
- Se ejecuta la creación de la gráfica de barras y se le muestra al usuario

Durante el tiempo en el que está ocurriendo esto al usuario se le muestra una ventana emergente que le informa de que se está recopilando la información solicitada.

Consulta por país

En este caso se utiliza la misma actividad que antes pero se configura la lista para que los valores que muestra sean los de los países del sistema en vez de los de las enfermedades. Una vez comprobados los campos rellenos y su validez se siguen estos pasos:

- Se crea una petición de tipo GET con el parser de JSON
- Se envía la primera fecha y el país a consultar
- Se llama al fichero PHP adecuado y se le pasan como parámetros la fecha y la enfermedad
- Se realiza la consulta en la base de datos que devuelve una lista con el número de afectados por cada enfermedad en valor acumulado hasta esa fecha
- Al llegar a la aplicación los datos se extraen de un objeto JSONArray y se almacenan en un array
- Se repite el proceso con la segunda fecha
- Se restan los valores de los arrays de enfermos para obtener el total de los enfermos entre esas dos fechas
- Se le pasa el resultado a la función generadora de el gráfico de tartas
- Se genera el gráfico y se muestra al usuario

Hay que mencionar que existen algunas variaciones respecto a si se utilizan para la consulta tweets con localización o todo el volumen de tweets. El primer caso es más sencillo porque sólo tiene que coger un grupo de valores en cambio el segundo tiene que coger los valores de tweets localizados y tweets sin localizar y sumarlos antes de almacenarlos en cada array.



Consulta del historial país/enfermedad

Este caso es algo distinto a los anteriores dado que en su pantalla de actividad se muestran tanto una lista de países como una lista de enfermedades de las que seleccionar. El resto de elementos se mantienen iguales. En esta ocasión cuando se realizan las consultas existen variaciones respecto a las peticiones anteriores:

- Se crea una petición de tipo GET con el parser de JSON
- Se envían las dos fechas juntas, el país y la enfermedad a consultar
- Se llama al fichero PHP correspondiente y se le pasan los parámetros.
- Se realiza la consulta en la base de datos que devuelve una con varios números. Estos números representan cada día y el valor de enfermos de ese día.
- Al llegar a la aplicación los datos se extraen de un objeto JSONArray y se almacenan en un array
- Se le pasa el array a la función generadora de la gráfica x y
- Se genera la gráfica y se muestra al usuario

Este es el funcionamiento básico de la aplicación en Android, más adelante se entrara en detalle de cómo se han implementado los pasos detallados anteriormente.

4.2.2 Comunicaciones

Las comunicaciones de la aplicación con el servidor se realizan mediante WI-FI o 3G por lo que en el manifest de la aplicación hay que activar los permisos de internet. La forma de realizar las consultas es a través de objetos JSON utilizando el servicio GET (POST no se usa porque desde la aplicación no se envía nada para almacenar, sólo se consulta). Además para la interpretación de las respuestas recibidas se utiliza un parser de JSON cuya misión es la de extraer la información útil del objeto recibido. Tanto el parser como los detalles de comunicación se especifican en la parte de “Desarrollo”.

4.2.3 Base datos

La base de datos en la aplicación tiene una funcionalidad muy simple. Almacena un valor de tipo booleano que al instalar la aplicación se establece como false. Una vez ejecutada la aplicación ese valor cambia a true y no vuelve a ser modificado.

4.2.4 Interfaz de usuario

En esta sección se pretende mostrar las pantallas de la aplicación así como explicar sus decisiones de diseño y enseñar los detalles que se han mencionado a lo largo de las anteriores secciones. Es la parte encargada de conectar al usuario principal con el sistema. Desde aquí se puede acceder a las distintas consultas y posteriores resultados. Es la única parte que el usuario podrá ver por lo que sus puntos fuertes deben ser el diseño y la usabilidad.

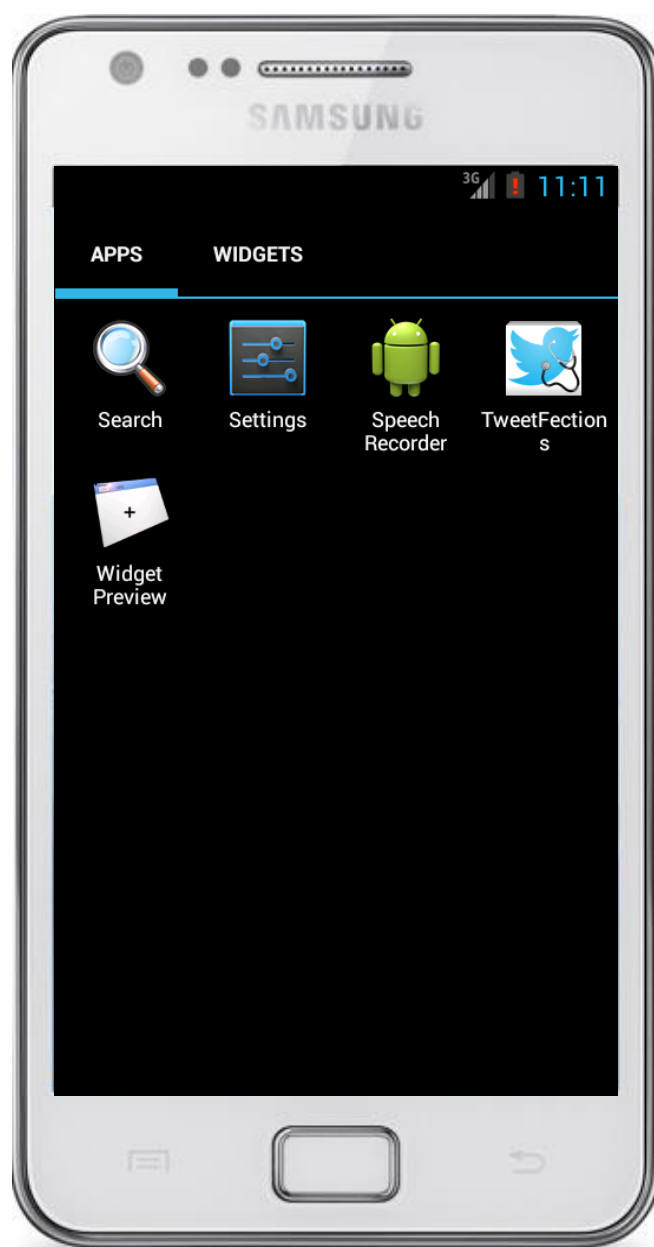


Ilustración 22 Pantalla Android 1

En esta ilustración se puede observar cómo aparece el icono con el logotipo de la aplicación en el menú de aplicaciones del teléfono. Al pulsarlo, por primera vez nos dirige a la siguiente pantalla.

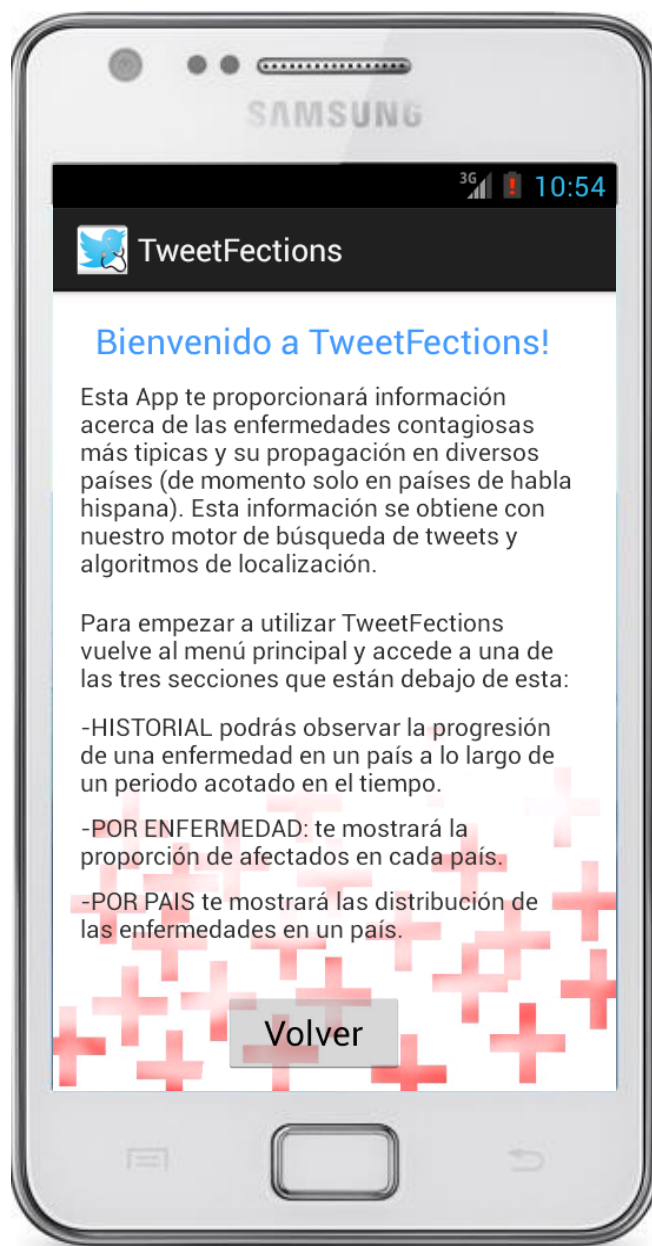


Ilustración 23 Pantalla Android 2

En esta captura se muestra el pequeño tutorial que se ejecuta al iniciarse la aplicación. Al hacer clic en volver nos conducirá directamente al menú principal. Como se puede observar la información es bastante clara y se utiliza un lenguaje cercano.

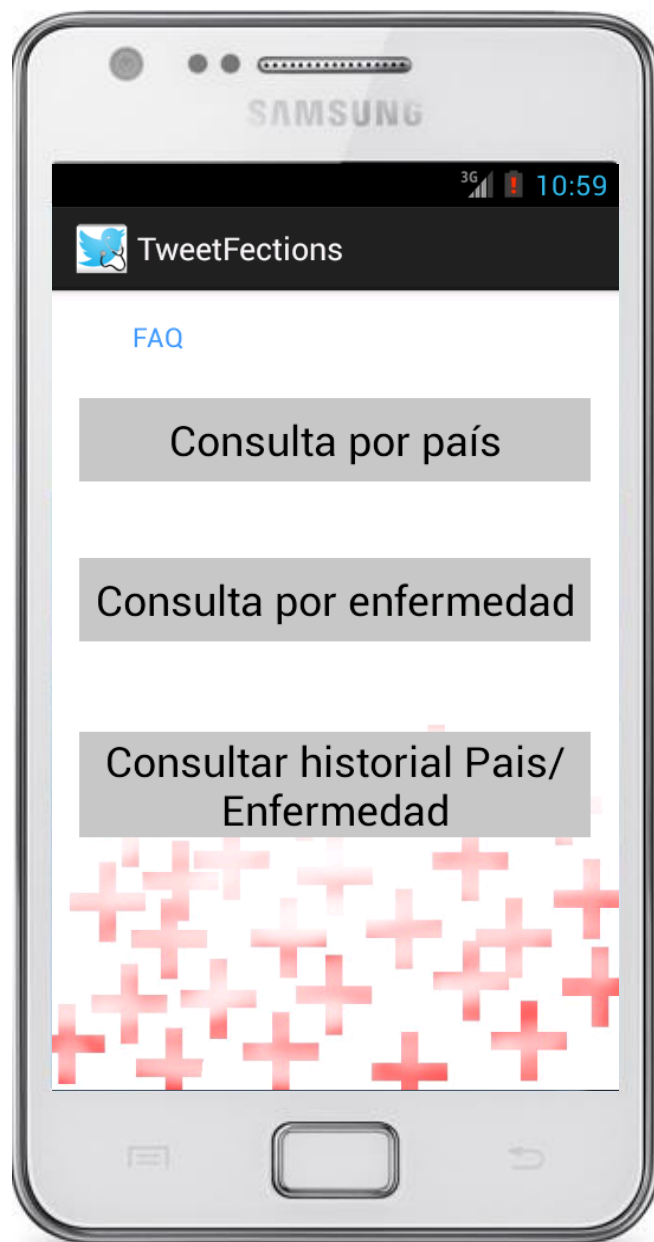


Ilustración 24 Pantalla Android 3

En esta pantalla se muestra el menú principal donde se puede elegir acceder al FAQ (esquina superior izquierda) o a las actividades de consulta. Pulsando sobre la primera se pasa a la siguiente captura de pantalla.



Ilustración 25 Pantalla Android 4

Al pulsar sobre consulta por país. Se pueden observar los elementos anteriormente descritos además tiene una serie de indicaciones para hacer más sencillo su uso. Al pulsar sobre la lista desplegable aparece una lista de los países seleccionables como se puede ver a continuación.



Ilustración 26 Pantalla Android 5

La lista contiene todos los países que se encuentran en el sistema. De momento sólo se han elegido los principales países de habla hispana con mayor índice de uso de Twitter. Una vez seleccionado uno hay que elegir la fecha.

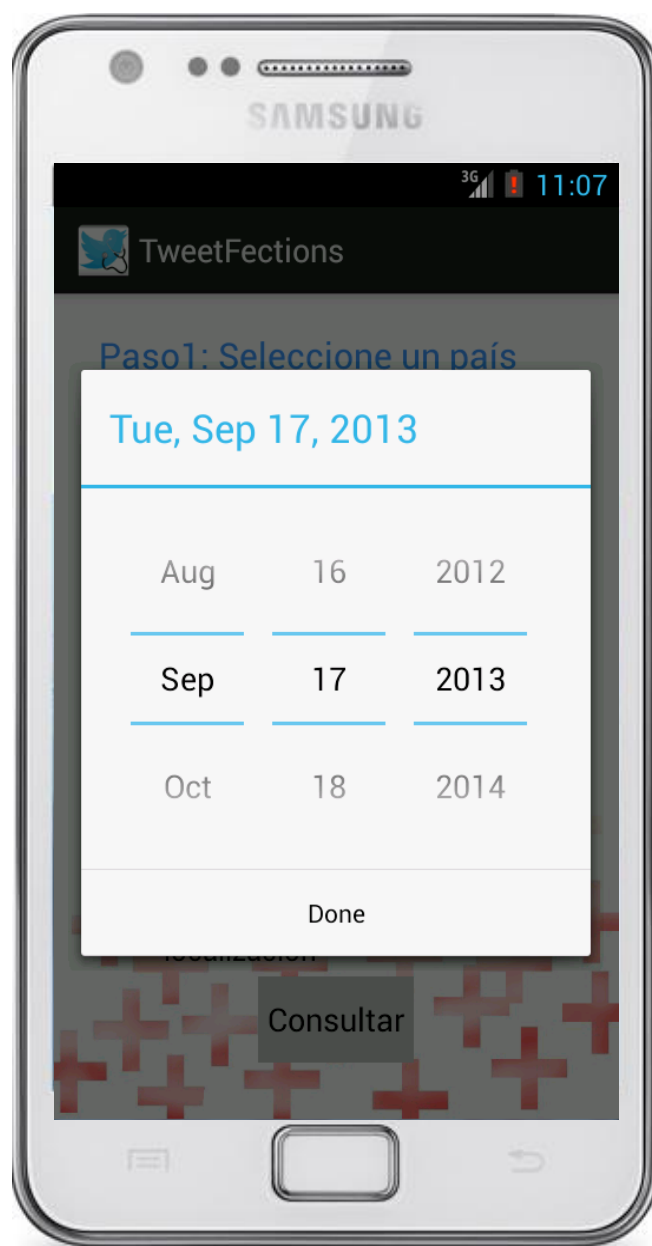


Ilustración 27 Pantalla Android 6

Al pulsar sobre seleccionar fecha de inicio o de fin aparece un calendario similar a este donde el usuario elegirá un par de fechas para configurar la consulta. Una vez los datos han sido rellenados se pulsa el botón “Consultar”.

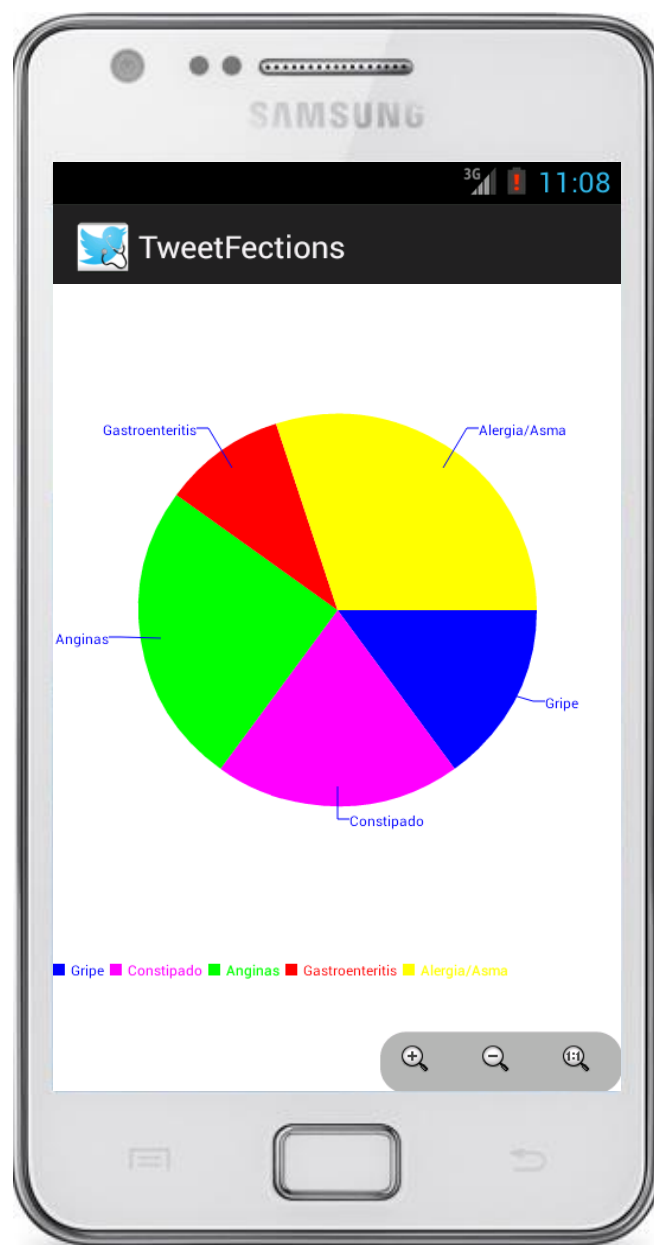


Ilustración 28 Pantalla Android 7

Al pulsar sobre el botón consultar en búsqueda por país se obtienen los resultados en un gráfico de tarta. Cada color representa una enfermedad y el tamaño de la porción representa el volumen de afectados relativos (de todos los afectados por las enfermedades de cuales hay mas).



Ilustración 29 Pantalla Android 8

Si en vez de pulsar consulta por país se pulsa en consulta por enfermedad se accede a esta pantalla. Como se puede observar la pantalla es similar a la de consulta por país es por ello que se utiliza la misma actividad para ambos dado que sólo hay que cambiar el contenido de la lista desplegable.



Ilustración 30 Pantalla Android 9

En esta pantalla se puede observar el detalle de la lista desplegable con el nuevo contenido. En esta ocasión muestra las 5 enfermedades sobre las que se puede realizar la consulta.



Ilustración 31 Pantalla Android 10

Si no se rellenan los campos de fecha la consulta no tiene lugar y se muestra al usuario un mensaje flotante indicándole el error. De la misma manera sucede si la fecha inicial es mayor que la fecha final o si la fecha final corresponde al día actual.



Ilustración 32 Pantalla Android 11

No se mencionó anteriormente pero al pulsar sobre la consulta aparece un texto flotante durante la recuperación de los datos del servidor. Gracias a esto el usuario sabe que la aplicación está trabajando. Si se hubiera dejado en blanco parecería que no ocurre nada

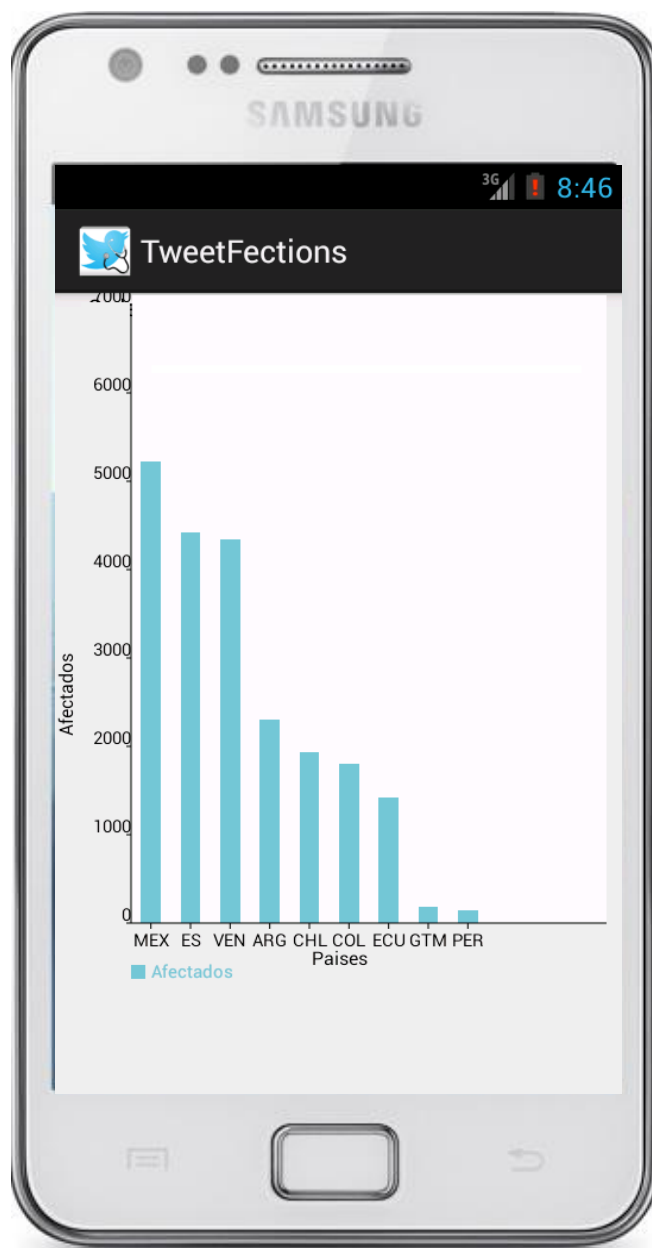


Ilustración 33 Pantalla Android 12

Tras pulsar sobre “Consultar” en la pantalla de consulta por enfermedad se mostrarán los resultados de forma similar a esta imagen. El eje y representa la cantidad de afectados y el eje x representa los países del sistema.

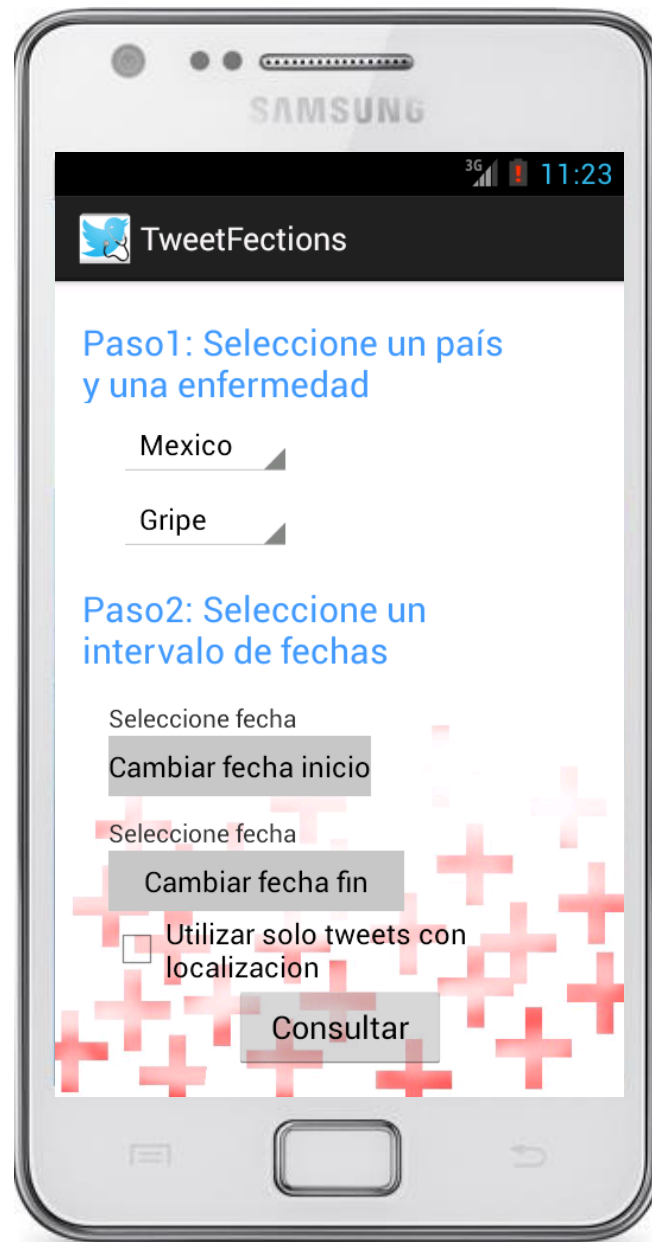


Ilustración 34 Pantalla Android 13

La última pantalla a la que se tiene acceso desde el menú principal es la de consulta del historial país/enfermedad. Como se mencionó anteriormente en esta pantalla se eligen tanto país como enfermedad. Por lo tanto como hay que incluir un elemento nuevo en la interfaz se desarrolla en una actividad propia.

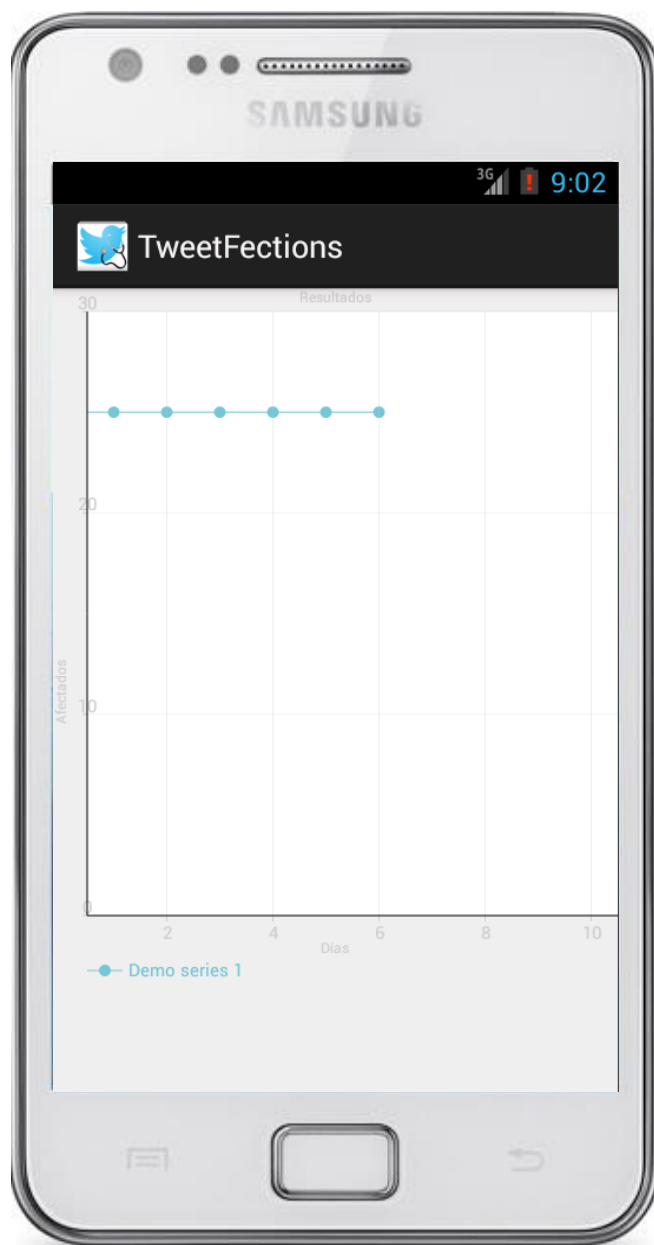


Ilustración 35 Pantalla Android 14

Al realizar la consulta por historial país/enfermedad se mostrará una gráfica similar a esta donde el eje y representa los afectados y el eje x los días de la consulta

Están son todas las pantallas de interfaz de usuario que tiene la aplicación se ha intentado mantener un diseño sencillo y funcional que mantenga los colores corporativos de Twitter pero con variaciones personales relacionadas con el tipo de servicio que se ofrece.

4.3 Servidor

La parte del servidor es la encargada de proporcionar los datos a la aplicación móvil cuando los necesite por eso debe ser robusta. Como se mencionó en la sección “Estado del Arte” para la configuración de todo el servidor se ha utilizado un software que agrupa todas las herramientas que se necesitan. Este es el paquete WAMP, cuya instalación es muy sencilla y se realiza siguiendo un asistente de instalación. Una vez desplegadas las herramientas llega el momento del diseño. La arquitectura ofrecida por WAMP y que sigue este servidor es:

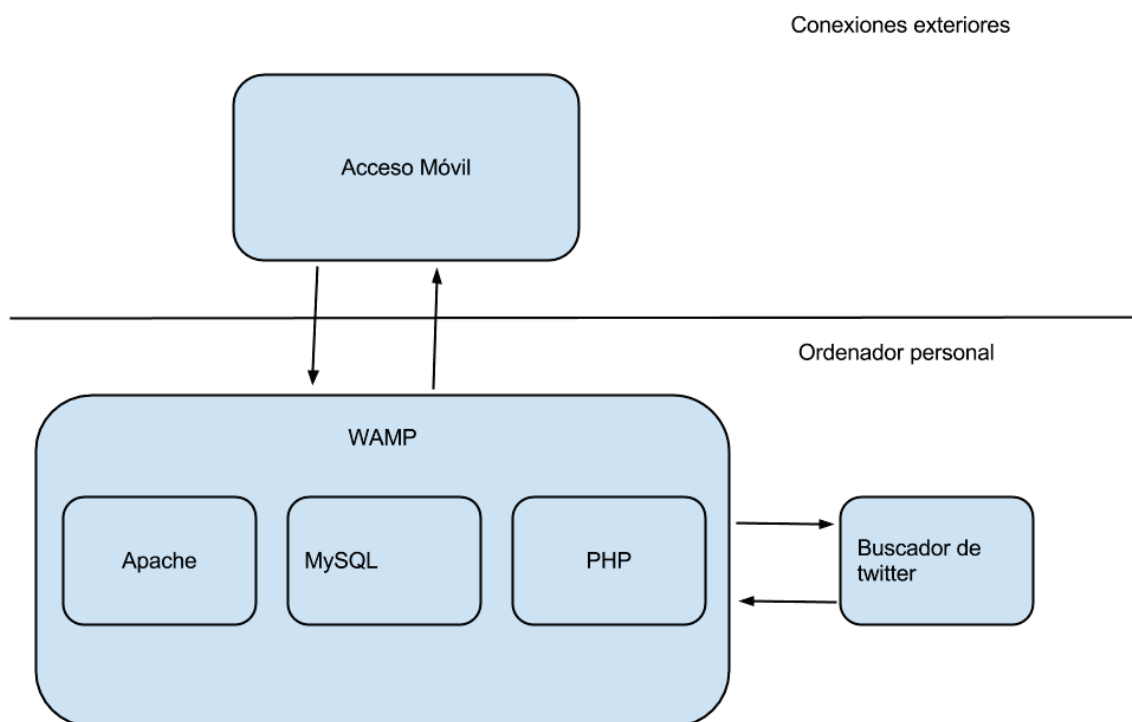


Ilustración 36 Arquitectura WAMP

Por lo tanto los elementos principales del diseño de esta parte del sistema son: Base de datos MySQL, servidor Apache, Ficheros PHP y buscador de Twitter.

4.3.1 Base de datos

En la base de datos se almacenan todos los datos que se han obtenido de las consultas en Twitter. Estos datos se almacenan siguiendo un modelo relacional que se explica más adelante. Los datos se almacenan y se piden a través de consultas MYSQL a través de los ficheros PHP del servidor.

Sigue un modelo relacional como el que se observa en la siguiente figura. El modelo no puede ser más simple, una única tabla.



Resultados
-Fecha
-País
-Enfermedad1
-Enfermedad2
-Enfermedad3
-Enfermedad4
-Enfermedad5
-Enfermedad1SL
-Enfermedad2SL
-Enfermedad3SL
-Enfermedad4SL
-Enfermedad5SL

Ilustración 37 Tabla base de datos

Como se puede observar hay algunos campos que parecen repetidos, pero no es así, simplemente unos son para datos recopilados de tweets con localización y otros para datos recopilados sin tweets de localización. Es importante mencionar que los valores de cada enfermedad se representan como un acumulado hasta la fecha. Esto se ha hecho así para optimizar el número de accesos que se realizan a la base de datos. Por ejemplo si una consulta desea saber cuáles son los porcentajes de las enfermedades para un país entre el 1 de enero y el 15 de enero para conseguir todos los datos habría que hacer 15 consultas a la base de datos (una por cada registro) en cambio con el modelo de datos acumulados sólo hace falta acceder a 2 registros, al de la fecha de inicio y al de la fecha de fin y restarlos entre ellos. Se podría optimizar también utilizando la función BETWEEN pero el volumen de datos devuelto sería demasiado grande (5 enfermedades por registro por 15 registros una lista de 75 registros) con este sistema el número de registros devueltos es siempre el mismo. Aunque este sistema sólo es válido para las consultas por país y por enfermedad. La consulta por historial sigue un proceso algo más complejo.

Los campos de la tabla son:

Fecha: se almacena la fecha del tweet recogido

País: Se almacena el país al que pertenecen los datos

Enfermedad: Estos campos hacen referencia a las distintas enfermedades y se almacena el número de tweets recogidos para cada enfermedad acumulados hasta la fecha de esa tabla. Enfermedad 1 es la gripe, enfermedad 2 es alergia/asma, enfermedad 3 es catarro/constipado/resfriado, enfermedad 4 es gastroenteritis/vómitos/diarrea y enfermedad 5 es faringitis/anginas.

Enfermedad SL: Estos campos siguen el mismo orden de correspondencia con enfermedad que los campos de Enfermedad pero tienen la peculiaridad de que sólo almacenan el volumen de los tweets sin localización.



Todos los datos de la base de datos son rellenados tras ser cribados y pasar los filtros del programa en java para consultas de twitter. Para el manejo de la base de datos se utilizarán los siguientes comandos:

SELECT: Sirve para realizar consultas, se puede configurar que datos se quieren y en función de qué condiciones deben estar los datos devueltos. Es el operador principal que se utiliza para realizar las consultas desde la base de datos

INSERT: Este operador lo utiliza únicamente el programa en Java que busca tweets. Sirve para crear nuevas tablas dentro de la base de datos. Se deben rellenar los parámetros que tienen esas tablas. Las inserciones son unas de las operaciones más abundantes en este sistema.

UPDATE: Es parecido a INSERT pero con la particularidad de que no crea una nueva tabla si no que actualiza los valores de una tabla ya existente. Este operador se utiliza conjuntamente a las inserciones de los datos por el programa en Java.

Para conseguir mantener el acumulado de afectados se realiza un proceso un tanto particular cada vez que se rellena la tabla por el programa en Java. Primero se hace un SELECT del registro del día anterior. Se suman esos datos a los generados por el buscador de Twitter y una vez sumados se realiza la inserción en un nuevo registro con la fecha del día actual.

4.3.2 Servidor Apache

Del servidor Apache no hay mucho que decir a nivel de diseño dado que no se ha implementado nada. Es un servicio que corre en el ordenador principal y que permite el acceso remoto. En un futuro se podría configurar una dirección DNS propia para ofrecer acceso desde cualquier parte del mundo, pero como tener una dirección propia cuesta dinero en este proyecto no se ha implementado y se ha simulado todo el funcionamiento a través de localhost. El funcionamiento es exactamente igual pero con la particularidad de que sólo se tiene acceso desde la propia red del ordenador en el que se ejecuta el servidor.

La parte de instalación y comprobación del funcionamiento se explicará en la parte de “Desarrollo” aunque para que se entienda mejor su lugar dentro del sistema se ha creado el siguiente diagrama.

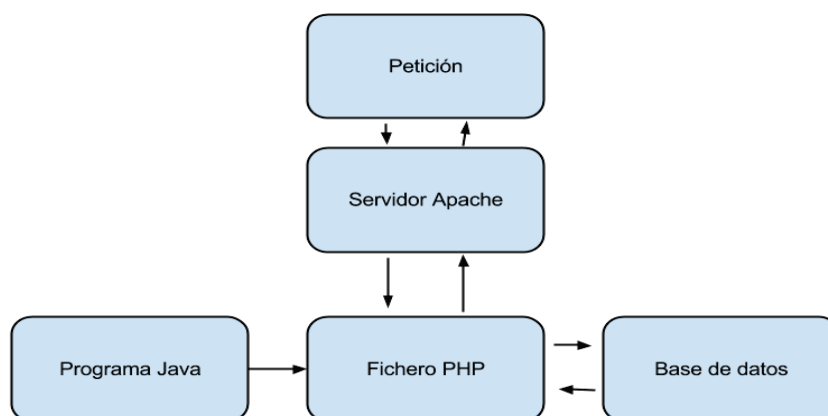


Ilustración 38 Diagrama sistema

Como se puede observar el servidor se encuentra justo en el camino de las peticiones externas y contiene en su directorio “www” una serie de fichero PHP que ofrecen los servicios a las peticiones. Estos ficheros y sus funciones se detallan a continuación.

4.3.2 Servicios y ficheros PHP

Los servicios ofrecidos por el servidor van en función de una serie de ficheros programados en PHP, el acceso a estos ficheros se realiza tanto desde la aplicación móvil como desde el buscador de tweets en Java por lo que son una parte indispensable del sistema. En el diseño del proyecto se han creado 7 ficheros PHP:

- **DBConfig.php:** en este fichero se detallan los campos necesarios para acceder a la base de datos del sistema.
- **DBConnect.php:** este fichero sirve para realizar la conexión a la base de datos es accedido tanto por las consultas como por las inserciones ya que siempre hay que conectarse a la base datos antes de realizar cualquier acción en ella.
- **Insertar.php:** Este fichero sirve para realizar los INSERT en la base de datos, es utilizado por el programa en Java para enviar los resultados y contiene una serie de métodos y condicionales que se explicarán más adelante
- **ConsultaPais.php:** este fichero ofrece el servicio para realizar las consultas por país desde la aplicación móvil



- **ConsultaEnfermedad.php:** este fichero ofrece el servicio para realizar las consultas por enfermedad desde la aplicación móvil
- **ConsultaHistorial.php:** este fichero ofrece el servicio para realizar las consultas por historial desde la aplicación móvil
- **ConsultaJava.php:** mediante este fichero se ofrece el servicio de consulta para el programa en Java.

Como se puede observar los 4 últimos son muy parecidos pero se ha decidido realizarlos de manera separada para evitar posibles conflictos. Si fuese un volumen mayor de ficheros sí se debería buscar una forma de optimizarlo.

4.3.3 Programa en Java

Esta es la piedra angular del sistema. Sin este programa no se podría haber realizado el proyecto y, aunque puede ser algo complejo de explicar, entre esta sección y la de “Desarrollo” quedará bastante claro cómo funciona.

A nivel general lo que hace el programa es conectarse a Twitter a través de su API utilizando una librería. Esta librería ofrece una serie de métodos para conseguir información de Twitter. Dado que con la nueva actualización de Twitter el volumen de consultas que se puede realizar está limitado a X número cada hora, el programa se ejecuta varias veces al día. Por esta razón en la aplicación móvil no se puede acceder al día actual puesto que los datos aún no están recopilados.

El programa contiene un array con 5 expresiones para buscar por cada enfermedad. Estas expresiones han sido extraídas tras realizar estudios en Twitter y valorar los resultados.

- **Gripe:** La gripe es una de las enfermedades más típicas de nuestros días, además es contagiosa, por tanto debe ser una de las candidatas a nuestro estudio. Las expresiones utilizadas para su búsqueda son:
 - Tengo gripe
 - Estoy con gripe
 - Tengo gripa
 - Estoy con gripa
 - He pillado la gripe
- **Alergia/Asma:** A pesar de no ser la misma enfermedad muchas veces están relacionadas y aunque no son enfermedades contagiosas se han incluido para que aquellas personas alérgicas al polen o al polvo puedan saber dónde hay más afectados. Las expresiones a utilizar en las búsquedas son:
 - Tengo alergia
 - Estoy con alergia
 - Tengo asma
 - Respirar asma
- **Gastroenteritis:** Es una enfermedad que se puede deber al consumo de algo en mal estado o al contagio. Los síntomas son diarrea y vómitos. Las expresiones a utilizar son:
 - Tengo gastroenteritis
 - Estoy con gastroenteritis
 - Tengo diarrea
 - Tengo vómitos
 - Estoy enfermo con vómitos



- **Faringitis/Anginas:** La faringitis y las anginas pueden no ser contagiosas pero suelen ser síntoma tanto de otras enfermedades como de posibles virus. Sus expresiones son
 - Me duele la garganta
 - Tengo faringitis
 - Me duelen las anginas
 - Tengo anginas
 - Estoy enfermo con anginas
- **Constipado/Resfriado:** similar a la gripe aunque por lo general sin fiebre. También es una enfermedad muy extendida sobre todo a nivel estacional por lo que no podía faltar en la lista. Los resultados se obtienen mediante las consultas por las siguientes expresiones:
 - Estoy constipado
 - Estoy acatarrado
 - Tengo catarro
 - Estoy resfriado
 - He pillado un catarro

El programa realiza las consultas a Twitter por cada enfermedad genera un fichero de texto en el que se almacenan todas las localizaciones de las consultas extraídas para esa enfermedad. Aquellos tweets sin localización incrementan un contador que se guarda en una variable definida para los tweets sin localización para una enfermedad. Después el programa busca en los ficheros, en los que se han guardado las localizaciones, las cadenas de texto correspondientes a los países. De momento sólo se ha implementado el nombre del país pero en un futuro se podría ampliar a ciudades y pueblos. Estos países se han elegido por su cantidad de cuentas, uso de Twitter y que su idioma es el español. Los países que conforman el sistema son: México", "España", "Venezuela", "Colombia", "Argentina", "Chile", "Perú", "Guatemala", "Ecuador". Por cada acierto se incrementa el contador de ese país para esa enfermedad.

Cuando se terminan este proceso se cuenta cuantos de los tweets con localización no tenían una localización que se ajustara a los países y ese volumen se suma a los tweets sin localización. Ese volumen de tweets sin localizar para cada enfermedad se reparte entre los países en función de su población y uso de Twitter. Para distribuir los tweets sin localización de la manera más justa razonable se utilizan una serie de coeficientes. Posteriormente se aplica un coeficiente de reducción tanto a tweets con localización como tweets sin localización.

Tras acabar los ajustes de los datos se realizan las inserciones en la base de datos tal y como se ha mencionado en la sección de base de datos.

En el proceso anterior se mencionan una serie de coeficientes. Los coeficientes son unos valores creados para el conjunto de los países. Se supone que se han registrado la mayoría de países hispanohablantes por lo que todo el reparto se debe realizar sobre los países del sistema. El principal justificante de los coeficientes es volumen de la población de cada país unido a la cantidad de cuentas que tiene ese país registradas en Twitter. Los coeficientes para cada país son:

- México: 20% de los tweets sin localización.
- España: 17% de los tweets sin localización.
- Venezuela: 15% de los tweets sin localización.
- Colombia: 14% de los tweets sin localización.
- Argentina: 12% de los tweets sin localización.
- Chile: 8% de los tweets sin localización.
- Perú: 6% de los tweets sin localización.



- Guatemala: 4% de los tweets sin localización.
- Ecuador: 4% de los tweets sin localización.

Estos porcentajes se han realizado de dos maneras. Una de ellas fue buscando en internet toda la información sobre cuentas creadas y países de habla hispana que usan Twitter y la otra recuperando varios archivos de localizaciones y analizándolos a mano uno a uno y apuntando datos. Esto no se llevó a cabo mediante un programa porque en muchos casos las localizaciones eran pueblos pequeños que había que buscar en Google y no se podían contemplar tantos pueblos en un sistema. Al menos en un sistema realizado por una sola persona en un periodo de tiempo corto.

Además de los citados anteriormente a todas las enfermedades para todos los países se les aplica un coeficiente de reducción del 12%. Esto se debe a que el sistema de búsqueda no es perfecto y entre el 10 y el 15% de los tweets que se recogen con las expresiones citadas anteriormente no implican el padecimiento de una enfermedad si no que utilizan la expresión en otro contexto. Por eso para intentar mantener un volumen de datos lo más realista posible se ha decidido reducir todos los volúmenes en un 12%. En algunos casos este coeficiente será demasiado o se quedará corto pero es un valor que tras varias comprobaciones consigue ajustar el resultado bastante bien.

Los datos mediante los cuales se han extraído los coeficientes así como el proceso seguido se detallan en evaluación de resultados y ajustes.

Este proceso no se puede ejecutar de una sola vez por lo que se ejecuta a lo largo de 4 horas para poder obtener el mayor número de tweets diario. Todos los tweets recogidos se limitan a la fecha actual por lo que la ejecución se debe realizar a última hora del día.



5 Desarrollo

En esta sección se explicará paso a paso el funcionamiento de los distintos componentes del sistema y su implementación. Para no incluir código que puede resultar confuso las partes que lo requieran se explicarán cómo pseudocódigo y otras con capturas de pantalla del propio código desarrollado.

5.1 Desarrollo del servidor

El desarrollo del servidor sigue lo descrito por el diseño:

- Instalación de WAMP
- Creación de la base de datos
- Creación del programa buscador de tweets en Java
- Creación de los ficheros PHP

Dentro de cada una de estas secciones se encuentran las distintas partes que componen cada elemento del servidor

5.1.1 Instalación de WAMP

La instalación de WAMP es muy sencilla y se realiza siguiendo los siguientes pasos:

- Descargar WAMP de la plataforma oficial eligiendo la versión (64 o 32 bits).
- Se ejecuta el asistente de instalación y se le da a “Next” en la primera pantalla cerrando antes los programas abiertos. Aquí aparecen los elementos que se van a instalar así como sus versiones.
- Aparecerá el acuerdo de licencia, se acepta y se pasa a la siguiente pantalla
- Se elige el directorio de instalación.
- Después se elige si se quieren crear iconos en el escritorio y menú de inicio.
- Finalmente se hace clic en “Install”. Durante el proceso se pedirá que se seleccione el navegador por defecto que usará WAMP.
- Después se pide que si se tiene un servidor SMTP se incluya con el email del servidor.
- Después aparecerá un mensaje para permitir el acceso de WAMP a la red. Se permite el acceso.
- Por último se pulsa el botón finish.

Para arrancar el servidor una vez instalado hay que cerrar todos los programas que pueda utilizar el puerto que utiliza WAMP (por defecto el puerto 80). Una vez cerrados los programas se ejecuta desde el escritorio o desde el menú de inicio y aparecerá un icono en la bandeja inferior derecha del sistema. Ahí se puede seleccionar el idioma y acceder a las distintas herramientas.

Para comprobar que está funcionando se abre un navegador y se escribe localhost en la barra de direcciones si el servidor está funcionando correctamente aparecerá una pantalla similar a esta:

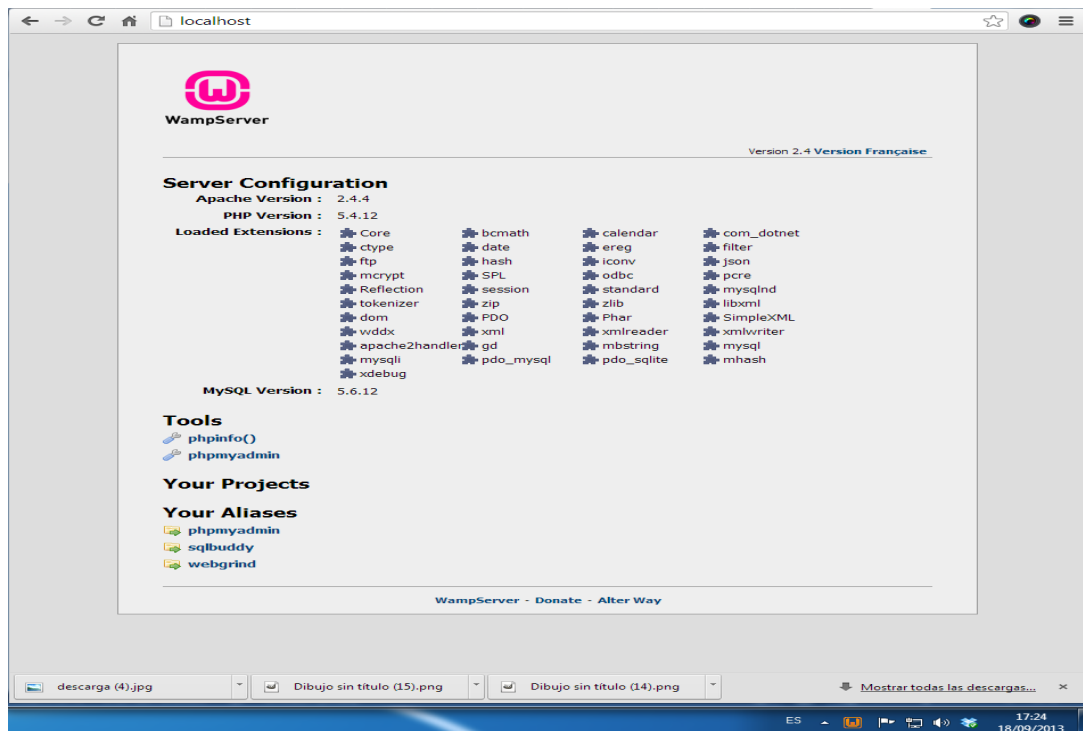


Ilustración 39 WAMP server

Una vez se tiene acceso a esa pantalla entonces el servidor está funcionando y se puede proceder a configurar su base de datos.

5.1.2 Creación de la base de datos

De cara a crear la base de datos se pueden elegir dos opciones la primera es crear la base de datos mediante sentencias SQL y la segunda crearla a través de phpMyAdmin, herramienta que proporciona WAMP para un manejo sencillo de las bases de datos. En el caso de este proyecto se ha elegido la segunda opción y los pasos que se han llevado a cabo para crear la base de datos son los siguientes:

- Descargar e instalar el programa de la página oficial de MySQL para crear bases de datos
- Crear la base de datos en el ordenador proporcionándole un usuario y una contraseña para su acceso
- Se escribe en la barra de direcciones del navegador <https://localhost/phpmyadmin/index.php>
- Se nos pedirá que introduzcamos la contraseña y el que hemos introducido al crear la base de datos

- Entonces tendremos acceso a una pantalla similar a esta, a la izquierda se pueden ver las bases de datos del sistema. La de este proyecto es tfgdb y la tabla es datos

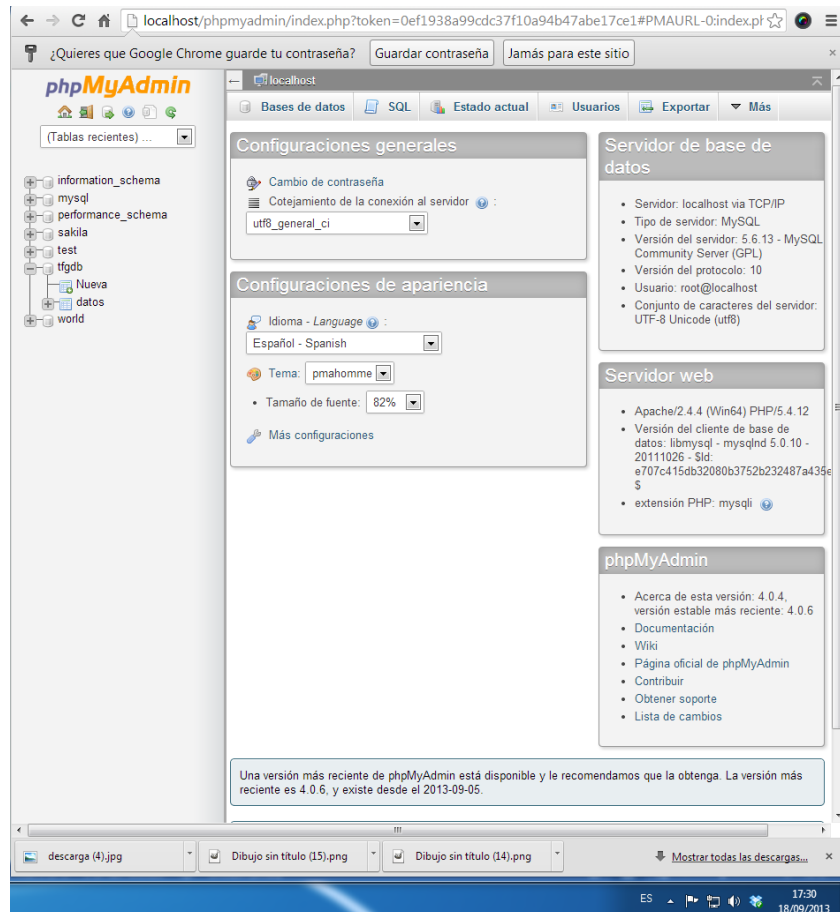


Ilustración 40 PHPMyAdmin 1

- Aunque en la imagen la tabla ya está creada hay que crearla a través de la interfaz de phpMyAdmin
- Lo primero es abrir la base de datos creada y después crear una tabla, para ello se especifica el nombre de la tabla y el número de columnas que tendrá. En este caso el número de columnas es 12 y el nombre de la tabla es datos

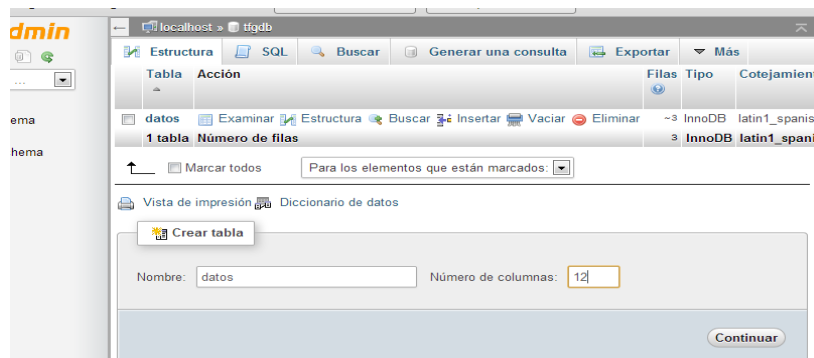


Ilustración 41 PHPMyAdmin 2

- Una vez creada la tabla se procede a rellenar cada columna con el nombre del campo y el tipo de valor que almacenará. Además se pueden añadir comentarios el tipo de motor de almacenamiento o el cotejamiento.

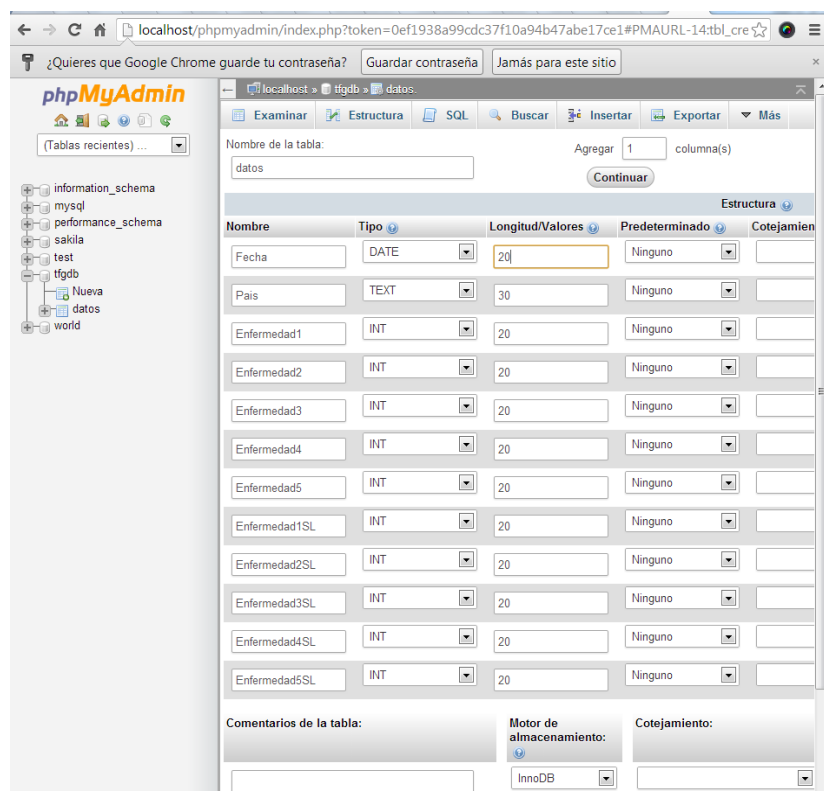


Ilustración 42 PHPMyAdmin 3

- Una vez terminado se hace clic en guardar y la tabla se crea en la base de datos. A partir de este punto ya se puede empezar a utilizar la base de datos

Los valores que se irán almacenando en los campos de Enfermedad X será siempre el acumulado del día anterior más los datos recogidos el día que indique la fecha.



5.1.3 Creación del programa buscador de tweets en Java

El programa como ya se ha mencionado varias veces es una parte muy importante del sistema. En esta sección se explicará su funcionamiento detallado y sus elementos.

Antes de empezar hay que mencionar los objetos que componen el proyecto en Java. Por un lado está implementada la clase JSON Parser encargada de extraer la información útil devuelta por el servidor tras las consultas o inserciones a la base de datos. Por otro lado, en una única clase está implementado el programa que busca los tweets y que los envía a la base de datos.

El programa en sí está compuesto por varios métodos. Estos métodos son:

- Main: es el método principal el primero que se ejecuta al iniciar la aplicación y desde donde se realizan las llamadas a los otros métodos.
- Buscar_Tweets: este método se encarga de buscar los tweets en Twitter y almacenar la información en distintos arrays
- Escribir_Fichero: este método sirve para escribir en los distintos ficheros utilizados como “cache” intermedia entre las búsquedas y las inserciones en la base de datos.
- Enviar_Datos: se ejecuta tras haber escrito todos los ficheros y se encarga de recopilar y ordenar la información de los ficheros y posteriormente enviar los datos a la base de datos.
- Twitter4j: Twitter4j es una librería que permite integrar la API de Twitter en Java. Se utiliza para acceder a Twitter y a los tweets desde el programa.

En el programa se utilizan varias estructuras de almacenamiento intermedio como pueden ser listas o arrays.

Main

El método main sigue el siguiente pseudocódigo (al final se expondrá un glosario con el nombre de cada variable y su función):

- Inicia el programa
- Se hace un for en función del tamaño del array que guarda las expresiones
 - Si contador de expresiones == 5
 - Se reinicia contador
 - Se cambia número de inserción
 - Se reinician contadores de tweets
 - Si contador de expresiones <5
 - Buscar_tweets parámetro expresión_enfermedad, número_inserción
- Bucle for en función del número de listasEnfermedadX
 - Escribir fichero parámetros listaEnfermedadX, contadorSLEnfermedadX, numero_inserción
- Enviar_datos
- Fin

Array expresiones: Es un array de tipo String donde se almacenan las expresiones que se van a buscar en Twitter.



Contador_expresiones: es un contador que indica cuantas expresiones se han buscado ya. Se incrementa tras terminar la búsqueda por esa expresión sirve para controlar qué enfermedad se está buscando.

Numero_insercion: es un número comprendido entre 0 y 4 e indica la enfermedad que se va a insertar en la base de datos o en qué lista se deben guardar las localizaciones

ContadorSLEnfermedadX: Es un contador para cada enfermedad que lleva la cuenta de aquellos tweets que no tienen localización. Se incrementa en el método buscar_tweets cada vez que la localización de un tweet es null.

Expresión_enfermedad: Es un String con una de las expresiones mencionadas en la fase de diseño.

listaEnfermedadX: es una lista por cada enfermedad en la que se almacenan en forma de Strings todas las localizaciones recopiladas durante la consulta de la enfermedad. Esta lista es la que se escribe en el fichero.

Buscar_tweets

El método buscar tweets utilizar la librería twitter4j y se integra en el método main. Su función es la de buscar tweets en función de las expresiones y almacenar las localizaciones obtenidas en sus correspondientes variables. El método funciona así:

- Busca por expresion_enfermedad recibida por parámetros
 - Mientras haya más datos en la query generada por la búsqueda
 - Si localización=null
 - contadorSLEnfermedadX (en función del parámetro número inserción) se incrementa en 1
 - Si no
 - Añadir elemento a listaEnfermedadX(pasada por parámetros) con localización
- Si contador_expresiones <5
 - Se vuelve a ejecutar con otra expresión
- Si no
 - Se duerme la ejecución durante 1 hora

Como se mencionó en el “Estado del Arte” la nueva API de Twitter restringe el número de consultas que pueden realizarse por hora por eso se consulta una enfermedad por hora. Esto se decidió tras comprobar que el máximo de tweets consultables se suele alcanzar cada 5 consultas de las expresiones de enfermedades.

Escribir_fichero

Este método recibe por parámetros las listas de las enfermedades, los contadoresSL y los números para su inserción en función de eso se escribe un fichero u otro. Su funcionamiento es el que sigue:

- Comprueba número_inserción
- Abre ficheroCLEnfermedadX adecuado
 - Mientras haya elementos en la listaEnfermedadX



- Añade cada elemento en una nueva fila
- Cierra fichero
- Abre ficheroSLEnfermedadX en función del número de inserción
 - Escribe el número de contadorSLEnfermedadX en fichero
- Cierra fichero

Esto se realiza por cada listaEnfermedadX que son 5. Por lo que al final se generan 10 ficheros 5 de ellos con localizaciones y cinco de ellos sin localizaciones.

FicheroCLEnfermedadX: que se genera por cada enfermedad y guarda las localizaciones obtenidas de los tweets de las consultas realizadas para esa enfermedad. En cada línea se escribe una localización.

FicheroSLEnfermedadX: similar al anterior pero solo se guarda el número de tweets sin localizar en una sola línea que contiene ese número.

Enviar_datos

Este es el último método que se ejecuta en el programa y tiene como finalidad enviar los datos a la base de datos y comprobar su inserción correcta. Además como deben guardarse los datos de manera acumulada se deben consultar los datos de la fecha anterior para sumarlos a los actuales. El funcionamiento de este método es el siguiente:

- Se recorren todos los ficheros con localización y se suman
- Bucle for por cada enfermedad
 - Bucle for por array_paises
 - Se abre fichero en función de primer bucle
 - Se busca cada país en el fichero con localización
 - Si se encuentra país
 - Se incrementa arrayPaisX en la posición de la enfermedad del fichero
 - Si no se pasa a siguiente línea
 - Al terminar los bucles se cuentan las líneas del fichero con localización
 - Se suman todos los datos de los arrayPaisX para esa enfermedad
 - Se resta el número de líneas a los datos sumados de arrayPaisX
 - Ese número se distribuye entre los arrayPaisX en función de los coeficientes de ajuste para tweets sin localización
 - Se cierra fichero
 - Se abre fichero sin localización
 - Se distribuye ese número entre los arrayPaisX en función de los coeficientes sumándolo a los registros que ya tuvieran
 - Se aplican coeficientes de reducción
- Bucle for por array_paises
 - Se obtiene fecha actual y se le resta un día
 - Se formatea la fecha como la de la base de datos
 - GET a base de datos pasando como parámetro el país y la fecha formateada



- Se extraen los datos de objeto JSON
- Se suma a cada posición de arrayPaisX el valor que le corresponde que se ha obtenido de la consulta
- POST a base de datos pasando como parámetros fecha actual, país, y el valor para cada enfermedad del arrayPaisX
- Si inserción correcta
 - Nada
- Si no
 - Mensaje de error
- Fin

Array_paises: Es un array que contiene en forma de cadena de texto los nombre de los países para los que se realizan las consultas.

ArrayPaisX: Es un array de 10 posiciones las 5 primeras son para número de tweets con localización para ese país para una enfermedad (Posición 0 gripe, posición1 alergia, posición 2 gastroenteritis, posición3 anginas, posición 4resfriado) las 5 últimas corresponden a las mismas enfermedades que las primeras pero solo de tweets sin localización.

GET: función web que se realiza mediante un fichero php alojado en el servidor sirve para pedir parámetros

POST: función web que se realiza mediante otro fichero php del servidor y sire para enviar información.

Fecha formateada: Como la base de datos tiene un formato de fecha determinado hay que formatear la fecha obtenida por java para que siga el mismo que el de la base de datos.

Coeficientes: Como se explica en la sección de “Diseño”, una gran cantidad de los tweets recogidos no tienen localización por lo que aquellos sin localización se reparten en función de la población de los países y su uso de Twitter según los coeficientes explicados en la parte de “Diseño”.

Por cada ejecución del programa completo se realizan 10 peticiones a la base de datos y 10 inserciones.

Meter gráfico de elementos del sistema

5.1.4 Comunicaciones

En este apartado se explica el desarrollo y funcionamiento de las comunicaciones que ofrece el servidor tanto a la aplicación móvil como al programa en Java. Las comunicaciones con la base de datos se ofrecen a través de servicios ofrecidos por ficheros PHP. El sistema, como se mencionó en la sección de diseño, está compuesto por:

- DBConfig.php
- DBConnect.php
- Insertar.php



- ConsultaPais.php
- ConsultaEnfermedad.php
- ConsultaHistorial.php
- ConsultaJava.php

De cada archivo se mostrará su parte más importante y se explicará su funcionamiento.

DBConfig.php

Este archivo configure los parámetros principales de la base de datos. Se usa simplemente como almacén de valores de variables y se importa en dbconnect.php

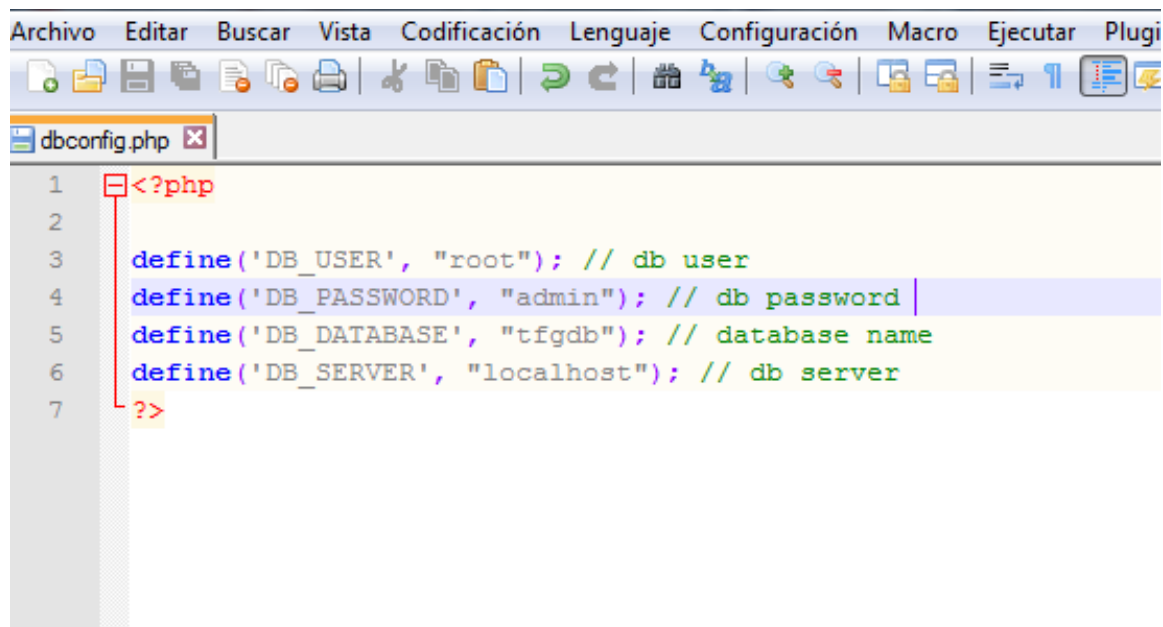


Ilustración 43 fichero PHP 1

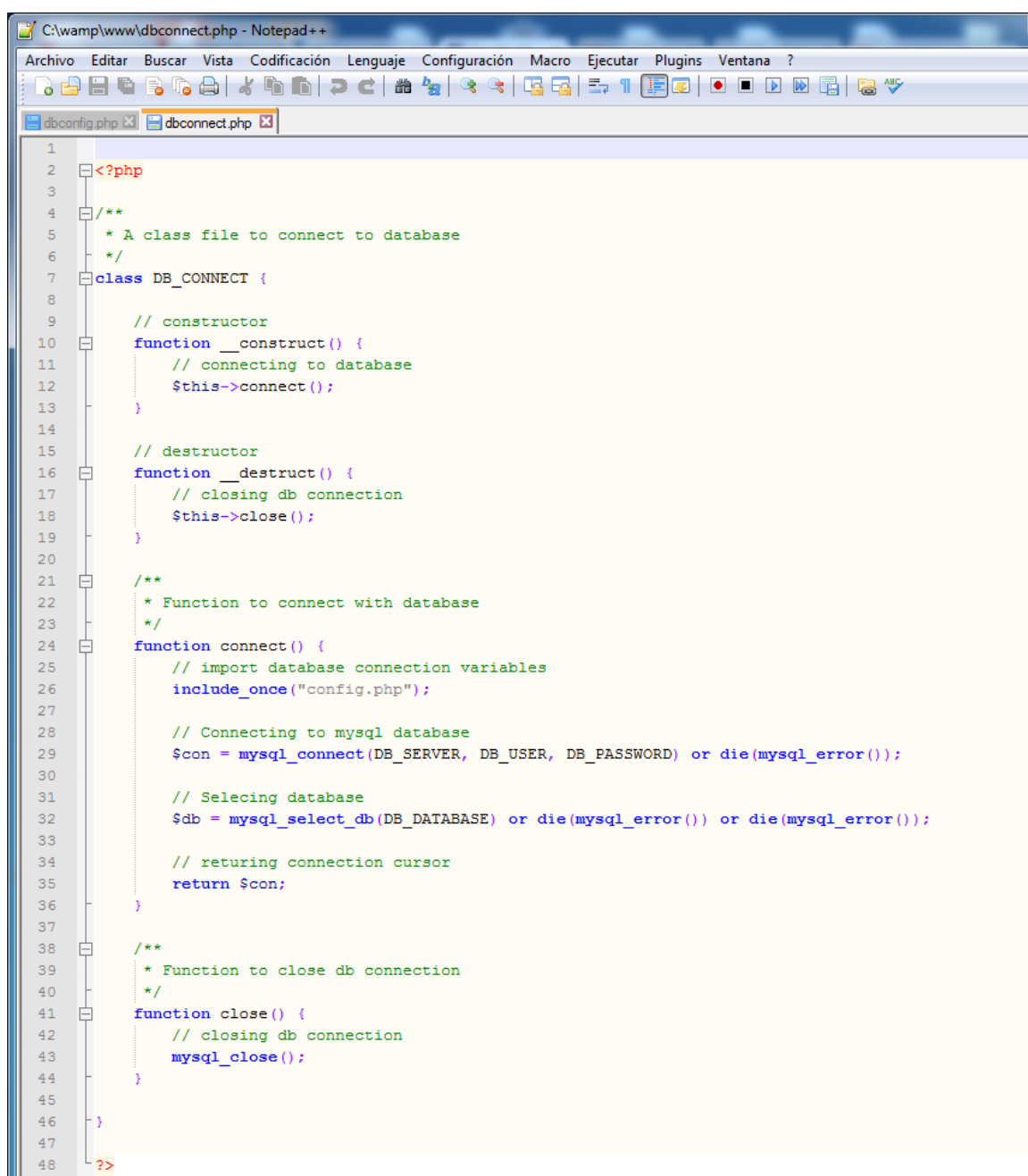
Como se puede apreciar en la imagen se definen 4 valores:

- DB_USER: es el usuario que se usa para acceder a la base de datos debe ser el mismo que el configurado durante la creación de la base de datos. En este caso su valor es "root".
- DB_PASSWORD: la contraseña del sistema para el usuario anterior su valor es "admin".
- DB_DATABASE: el nombre de la base de datos a la que se quiere acceder, en el caso del proyecto la base de datos se llama "tfgdb".
- DB_SERVER: dirección en la que se aloja la base de datos, como el sistema a día de hoy funciona sólo en red local su valor el "localhost"

DBConnect.php

Este fichero se encarga de crear la conexión a la base de datos. En la imagen inferior se puede observar que está formado por varias funciones.

- `__construct()`: llama la función `connect()`.
- `__destruct()`: llama a la función `close()`.
- `Connect()`: importa el fichero `config.php` y crea dos objetos uno de tipo conexión con la base de datos a la cual se conecta utilizando los valores de las variables del fichero `config.php` y otro objeto de tipo `db` donde se selecciona la base de datos. Finalmente se devuelve el cursor de la conexión a la base de datos.
- `Close()`: esta función simplemente cierra la conexión con la base de datos.



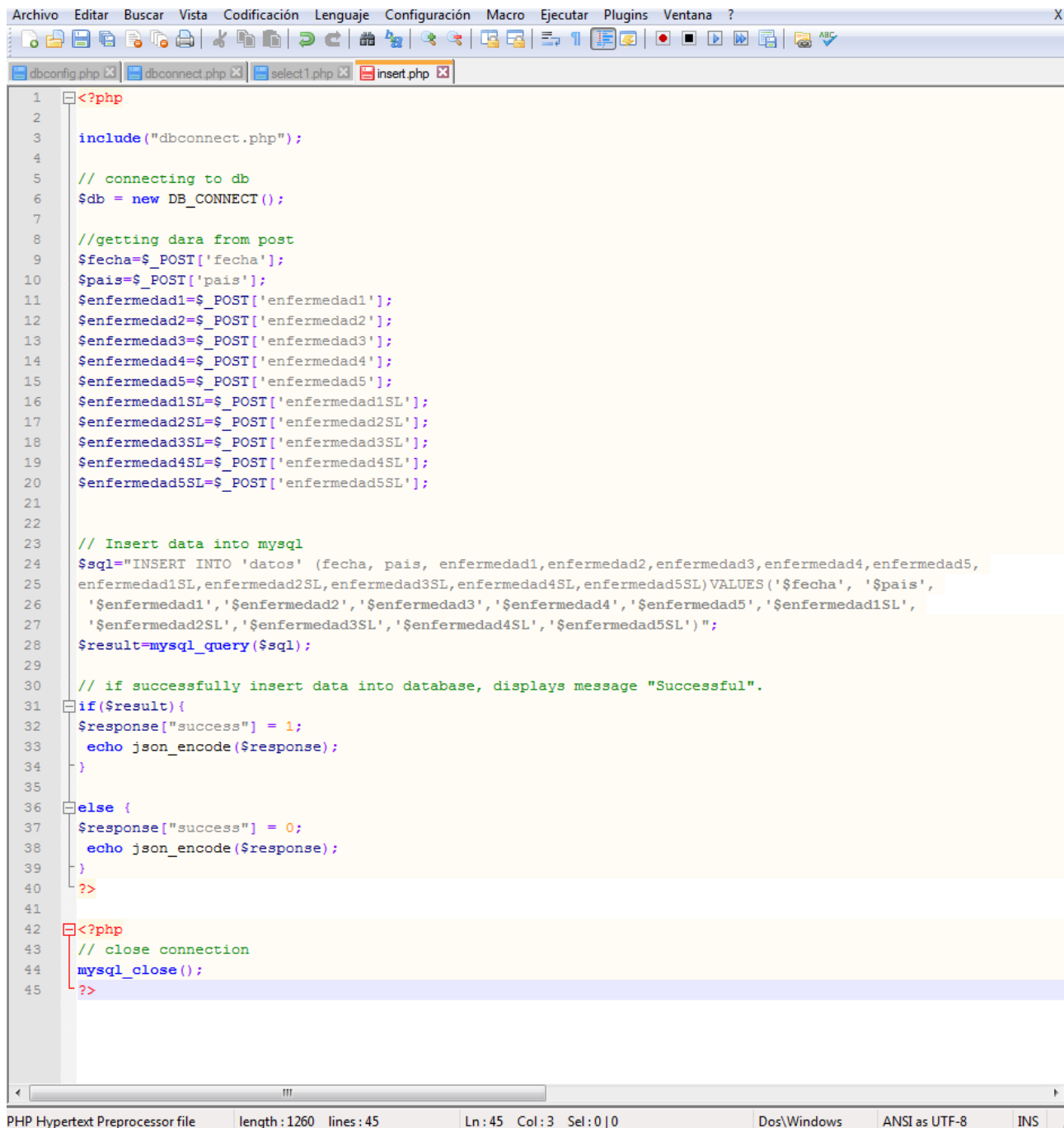
```
1
2 <?php
3
4 /**
5  * A class file to connect to database
6  */
7 class DB_CONNECT {
8
9     // constructor
10    function __construct() {
11        // connecting to database
12        $this->connect();
13    }
14
15    // destructor
16    function __destruct() {
17        // closing db connection
18        $this->close();
19    }
20
21    /**
22     * Function to connect with database
23     */
24    function connect() {
25        // import database connection variables
26        include_once("config.php");
27
28        // Connecting to mysql database
29        $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or die(mysql_error());
30
31        // Seleccioning database
32        $db = mysql_select_db(DB_DATABASE) or die(mysql_error()) or die(mysql_error());
33
34        // returning connection cursor
35        return $con;
36    }
37
38    /**
39     * Function to close db connection
40     */
41    function close() {
42        // closing db connection
43        mysql_close();
44    }
45
46 }
47
48 ?>
```

Ilustración 44 Fichero PHP 2



Insertar.php

Este fichero lo utiliza el programa en Java sirve para insertar en la base datos los datos recogidos de Twitter. Es un fichero muy simple que contiene una serie de variables donde se almacenan los datos incluidos en el POST hecho desde el programa, después se realiza la inserción y se comprueba el resultado devolviéndose un 1 en caso de inserción correcta y un 0 en caso de inserción incorrecta. En la imagen inferior se puede ver el fichero:



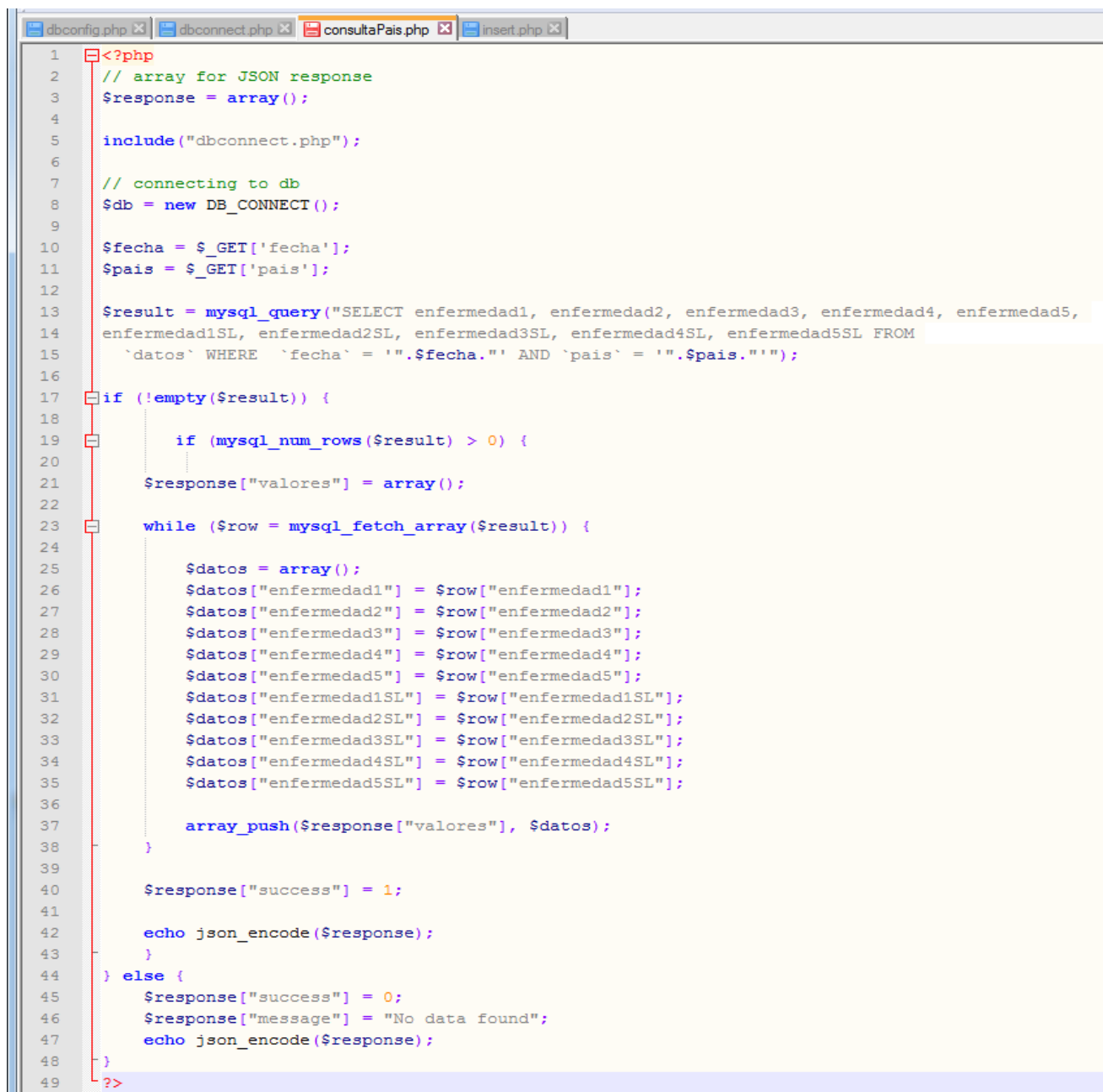
```
1 <?php
2
3 include("dbconnect.php");
4
5 // connecting to db
6 $db = new DB_CONNECT();
7
8 //getting data from post
9 $fecha=$_POST['fecha'];
10 $pais=$_POST['pais'];
11 $enfermedad1=$_POST['enfermedad1'];
12 $enfermedad2=$_POST['enfermedad2'];
13 $enfermedad3=$_POST['enfermedad3'];
14 $enfermedad4=$_POST['enfermedad4'];
15 $enfermedad5=$_POST['enfermedad5'];
16 $enfermedad1SL=$_POST['enfermedad1SL'];
17 $enfermedad2SL=$_POST['enfermedad2SL'];
18 $enfermedad3SL=$_POST['enfermedad3SL'];
19 $enfermedad4SL=$_POST['enfermedad4SL'];
20 $enfermedad5SL=$_POST['enfermedad5SL'];
21
22
23 // Insert data into mysql
24 $sql="INSERT INTO 'datos' (fecha, pais, enfermedad1,enfermedad2,enfermedad3,enfermedad4,enfermedad5,
25 enfermedad1SL,enfermedad2SL,enfermedad3SL,enfermedad4SL,enfermedad5SL) VALUES ('$fecha', '$pais',
26 '$enfermedad1', '$enfermedad2', '$enfermedad3', '$enfermedad4', '$enfermedad5', '$enfermedad1SL',
27 '$enfermedad2SL', '$enfermedad3SL', '$enfermedad4SL', '$enfermedad5SL') ";
28 $result=mysql_query($sql);
29
30 // if successfully insert data into database, displays message "Successful".
31 if($result){
32     $response["success"] = 1;
33     echo json_encode($response);
34 }
35
36 else {
37     $response["success"] = 0;
38     echo json_encode($response);
39 }
40 ?>
41
42 <?php
43 // close connection
44 mysql_close();
45 ?>
```

PHP Hypertext Preprocessor file length : 1260 lines : 45 Ln : 45 Col : 3 Sel : 0 | 0 Dos\Windows ANSI as UTF-8 INS

Ilustración 45 Fichero PHP 3

ConsultaPais.php y consultaJava.php

Este fichero se utiliza desde la aplicación móvil cuando se realiza una consulta por país. El fichero sigue exactamente el mismo modelo que consultaJava.php por eso se explican de manera conjunta.



```
1 <?php
2 // array for JSON response
3 $response = array();
4
5 include("dbconnect.php");
6
7 // connecting to db
8 $db = new DB_CONNECT();
9
10 $fecha = $_GET['fecha'];
11 $pais = $_GET['pais'];
12
13 $result = mysql_query("SELECT enfermedad1, enfermedad2, enfermedad3, enfermedad4, enfermedad5,
14 enfermedad1SL, enfermedad2SL, enfermedad3SL, enfermedad4SL, enfermedad5SL FROM
15 `datos` WHERE `fecha` = '". $fecha. "' AND `pais` = '". $pais. "'");
16
17 if (!empty($result)) {
18     if (mysql_num_rows($result) > 0) {
19         $response["valores"] = array();
20
21         while ($row = mysql_fetch_array($result)) {
22             $datos = array();
23             $datos["enfermedad1"] = $row["enfermedad1"];
24             $datos["enfermedad2"] = $row["enfermedad2"];
25             $datos["enfermedad3"] = $row["enfermedad3"];
26             $datos["enfermedad4"] = $row["enfermedad4"];
27             $datos["enfermedad5"] = $row["enfermedad5"];
28             $datos["enfermedad1SL"] = $row["enfermedad1SL"];
29             $datos["enfermedad2SL"] = $row["enfermedad2SL"];
30             $datos["enfermedad3SL"] = $row["enfermedad3SL"];
31             $datos["enfermedad4SL"] = $row["enfermedad4SL"];
32             $datos["enfermedad5SL"] = $row["enfermedad5SL"];
33
34             array_push($response["valores"], $datos);
35         }
36
37         $response["success"] = 1;
38
39         echo json_encode($response);
40     } else {
41         $response["success"] = 0;
42         $response["message"] = "No data found";
43         echo json_encode($response);
44     }
45 }
46
47 ?>
```

Ilustración 46 Fichero PHP 5

Como se puede observar en la imagen se seleccionan de la base de datos todos los valores de tweets de enfermedades tanto con localización como sin localización para la fecha y el país especificados en el GET. Si el resultado es positivo se procede a rellenar que el array será devuelto al cliente con los datos y se pone un 1 al campo success para informar de que la consulta ha tenido éxito. En caso contrario se pondría un 0.

ConsultaEnfermedad.php

Este fichero es muy parecido al anterior y la única variación que existe son los datos que se reciben para realizar el SELECT y los datos que se devuelven al cliente.



```
1 <?php
2 // array for JSON response
3 $response = array();
4
5 include("dbconnect.php");
6
7 // connecting to db
8 $db = new DB_CONNECT();
9
10 $fecha = $_GET['fecha'];
11 $enfermedad = $_GET['enfermedad'];
12
13 $result = mysql_query("SELECT ' ".$enfermedad."', pais FROM
14 `datos` WHERE `fecha` = ' ".$fecha." '");
15
16 if (!empty($result)) {
17
18     if (mysql_num_rows($result) > 0) {
19
20         $response["valores"] = array();
21
22         while ($row = mysql_fetch_array($result)) {
23
24             $datos = array();
25             $datos["pais"] = $row["pais"];
26             $datos["numero"] = $row["enfermedad1"];
27
28             array_push($response["valores"], $datos);
29         }
30
31         $response["success"] = 1;
32
33         echo json_encode($response);
34     }
35 } else {
36     $response["success"] = 0;
37     $response["message"] = "No data found";
38     echo json_encode($response);
39 }
```

Ilustración 47 Fichero PHP 6

En la imagen se aprecia que los parámetros pasados por el cliente para la consulta son el nombre de la enfermedad y una fecha. Y los datos devueltos es un array de países y números. El resto funciona de la misma manera que el anterior fichero.

[ConsultaHistorial.php](#)

```
1  <?php
2  // array for JSON response
3  $response = array();
4
5  include("dbconnect.php");
6
7  // connecting to db
8  $db = new DB_CONNECT();
9
10 $fecha1 = $_GET['fecha1'];
11 $fecha2 = $_GET['fecha2'];
12 $pais=$_GET['pais'];
13 $enfermedad = $_GET['enfermedad'];
14
15 $result = mysql_query("SELECT ".$enfermedad.", pais FROM
16 `datos` WHERE `pais`='".$pais."' AND `fecha` BETWEEN '".$fecha1."' AND '".$fecha2."' ");
17
18 if (!empty($result)) {
19
20     if (mysql_num_rows($result) > 0) {
21
22         $response["valores"] = array();
23
24         while ($row = mysql_fetch_array($result)) {
25
26             $datos = array();
27             $datos["numero"] = $row["enfermedad1"];
28
29             array_push($response["valores"], $datos);
30         }
31
32         $response["success"] = 1;
33
34         echo json_encode($response);
35     }
36 } else {
37     $response["success"] = 0;
38     $response["message"] = "No data found";
39     echo json_encode($response);
40 }
41 ?>
```

Ilustración 48 Fichero PHP 7

En este caso esta consulta la utiliza la consulta por historial país/enfermedad. Como se puede observar se le pasan los datos fecha1 (fecha inicial) fecha2 (fecha final), país y enfermedad. Y se devuelve un array de números, cada número se corresponde con una fecha. El resto de las funciones son similares a las de los ficheros anteriores.



5.2 Desarrollo de la aplicación

En esta sección se explica el desarrollo de la aplicación móvil. Los elementos principales de la aplicación son:

- Las 4 actividades descritas en el diseño
- Los 4 layouts para las actividades
- El manifiesto de la aplicación

A continuación se procede a describir cada una de las 4 partes listadas, su funcionamiento y su función en el sistema

5.2.1 Actividades

En los proyectos las funcionalidades generales se suelen distribuir en actividades. En este caso las 4 actividades que conforman el proyecto son:

- MainActivity.java: Clase principal que se ejecuta al iniciar la aplicación contiene botones que enlazan con las otras actividades
- FAQ.java: actividad que contiene la información de uso de la aplicación contiene un botón para volver al menú principal
- Historial.java: actividad que permite al usuario realizar las consultas por historial. Contiene los métodos para generar las gráficas y diversos elementos interactivos.
- ConsultasBasicas.java: actividad que permite al usuario realizar las consultas tanto por país como por enfermedad. Contiene los métodos para generar las gráficas y diversos elementos interactivos.

Todas las actividades cargan su layouts en el método onCreate

MainActivity.java

Esta actividad está compuesta por 3 botones que dan acceso a las consultas y un texto pulsable para acceder al FAQ. Para esta actividad está definida una lógica que se describe a continuación y que se ejecuta nada más iniciarse la actividad.

- Si existe base de datos
 - Abrir base de datos y comprobar valor primeraVez
 - Si valor primeraVez = false
 - Valor primeraVez = true
 - Se inicia actividad FAQ
 - Si no
 - No hacer nada
- Si no existe
 - Crear base de datos

Primera vez: es un valor de tipo booleano recogido del booleano de la base de datos..

La creación de la base de datos se realiza mediante las herramientas que incorpora Android para ello. La funcionalidad se llama SQLiteDatabase y nos permite definir y crea una base de datos contenida dentro del dispositivo. La base de datos de la aplicación móvil contiene un único valor de tipo booleano que en la creación de la base de datos se inicia con el valor false.

Además de esta lógica en esta actividad se configuran los eventos para las pulsaciones de los botones de la siguiente forma:

```
TextView tt= (TextView)findViewById(R.id.click);
Button bt1 = (Button) findViewById(R.id.buttonPais);
Button bt2 = (Button) findViewById(R.id.ButtonEnfermedad);
Button bt3 = (Button) findViewById(R.id.historial);

tt.setOnClickListener(new OnClickListener (){
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, FAQ.class);
        startActivity(intent);
    }
});

bt1.setOnClickListener(new OnClickListener (){
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, ConsultasBasicas.class);
        intent.putExtra("modo", true);
        startActivity(intent);
    }
});

bt2.setOnClickListener(new OnClickListener (){
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, ConsultasBasicas.class);
        intent.putExtra("modo", false);
        startActivity(intent);
    }
});

bt3.setOnClickListener(new OnClickListener (){
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, Historial.class);
        startActivity(intent);
    }
});
```

Ilustración 49 Código 1

Como se puede apreciar, al pulsar sobre los distintos botones (o texto pulsable) se inicia la actividad indicada para cada botón. Un detalle a remarcar es que cuando se pulsan los botones consulta por país o consulta por enfermedad en el intent que se genera para iniciar la actividad se incluye una variable llamada modo a true o a false en función del botón pulsado. Esto sirve para decir a la actividad común que comparten ambas consultas qué cargar en la pantalla y qué tipo de consulta se quiere realizar.



FAQ.java

Esta actividad no implementa nada más que un configurador de botón para volver al menú principal. La información incluida en la pantalla está a cargo de su layout. La configuración del botón se realiza como en la imagen anterior, es decir se crea un `onClickListener` que inicia la actividad `MainActivity`

Historial.java

En esta actividad se proporcionan al usuario las funcionalidades necesarias para realizar las consultas por historial país/enfermedad. La actividad incorpora varios elementos además de la lógica de funcionamiento.

- Elementos: estos elementos son los definidos en el layout. En la actividad se configura su funcionamiento y se implementan algunas funcionalidades o modos. Los elementos son:
 - Botón consultar: Este botón se pulsa para realizar la consulta, en su disparador se encuentra toda la lógica y las acciones que realizará la actividad para solicitar los datos a la base de datos.
 - Botones seleccionar fecha: pulsando estos botones se ejecuta el calendario interactivo.
 - Calendario: Historial.java cuenta con dos calendarios a través de los que definir las fechas de inicio y de fin. Se implementan mediante la clase `DatePickerFragment` y en ella se configura la forma en la que se mostraran las fechas o cómo se recogerán y formatearan una vez introducidas.
 - Campos de texto: la actividad también contiene dos campos de texto que se editan al seleccionar las fechas y muestran al usuario qué fechas se han elegido.
 - Generador de gráfica x y: El generador de la gráfica x y son dos clases de la librería `achartengine-1.1-0`. Esta librería permite generar diversos tipos de gráficas mediante el paso de unos parámetros.
 - Checkbox: la actividad cuenta también con un `checkbox` cuyo valor se comprueba a la hora de realizar las consultas y sirve para indicar si la consulta se realizará sólo con tweets con localización o si se utilizarán todos.
 - Listas interactivas: se implementan también dos listas interactivas una con todas las enfermedades y otra con todos los países. Al principio de la actividad se configura qué valores mostrará cada una.
- Lógica de funcionamiento: al pulsar el botón consultar se llevan a cabo una serie de procesos que se definen a continuación:
 - Si `fecha1==null` o `fecha 2==null`
 - Mostrar mensaje rellenar fechas
 - `Variable Ok=false`
 - Si `fecha 1>fecha 2`
 - Mostrar mensaje fecha inicio mayor fecha fin
 - `Variable OK =false`
 - Si `fecha2` o `fecha 1 == fecha actual`
 - Mostrar mensaje no se puede acceder a día actual
 - `Variable OK=false`
 - Si `fecha1==fecha2`
 - Mostrar mensaje seleccione distintas fechas



- Variable OK=false
- Si variable OK == false
 - Nada
- Si no iniciar actividad asíncrona
 - Ejecutar dialogo de cargando datos
 - Recoger datos usuario
 - Añadir datos a parámetros de consulta
 - GET a base de datos
 - Si respuesta = 1
 - Si checkbox seleccionado
 - Nada
 - Si no
 - Sumar localizados con sin localizar
 - Pasar parámetros a generador de gráficas
 - Terminar dialogo cargando datos
 - Mostrar gráfica
 - Si no
 - Terminar dialogo cargando datos
 - Mostrar mensaje no hay datos para esas fechas

A continuación se detallan algunos de los términos que pueden resultar confusos para el lector

Fecha1: fecha de inicio de la consulta

Fecha2: fecha de fin de la consulta

Mostrar mensaje: Se refiere a mostrar un mensaje flotante con distintos contenidos

Variable OK: variable que sirve para controlar que los parámetros introducidos son correctos

Actividad asíncrona: Android desde su versión 4 obliga a realizar todas las tareas pesadas mediante tareas asíncronas para no bloquear la interfaz del usuario

Dialogo cargando datos: dialogo que se muestra durante la petición de los datos al servidor

GET: llamada al fichero php para realizar la consulta en el servidor.



ConsultasBasicas.java

Esta actividad permite al usuario realizar las consultas por país o por enfermedad. Contiene elementos similares a los de historial.java pero su lógica de funcionamiento es muy distinta debido al carácter de las consultas realizadas.

- Elementos (los elementos que son iguales a los de historial.java no se explican)
 - Botón consultar
 - Botones seleccionar fecha
 - Calendario
 - Campos de texto
 - achartengine-1.1-0.
 - Checkbox:
 - Lista interactiva: en este caso sólo se muestra una lista interactiva. El contenido de esta lista depende de un valor pasado por MainActivity.java
- Lógica de funcionamiento
 - Si modo (recibido en el intent de MainActivity.java) == true
 - Contenido lista = enfermedades
 - Tipo de consulta=1
 - Si no
 - Contenido lista = países
 - Tipo de consulta=2
 - Si fecha1==null o fecha 2==null
 - Mostrar mensaje rellenar fechas
 - Variable Ok=false
 - Si fecha 1>fecha 2
 - Mostrar mensaje fecha inicio mayor fecha fin
 - Variable OK =false
 - Si fecha2 o fecha 1 == fecha actual
 - Mostrar mensaje no se puede acceder a día actual
 - Variable OK=false
 - Si fecha1==fecha2
 - Mostrar mensaje seleccione distintas fechas
 - Variable OK=false
 - Si variable OK = false
 - Nada
 - Si no iniciar actividad asíncrona
 - Si tipo de consulta = 1
 - Ejecutar consulta1
 - Si no
 - Ejecutar consulta2
- Consulta1
 - Ejecutar dialogo de cargando datos
 - Recoger datos usuario
 - Añadir datos a parámetros de consulta
 - GET a base de datos por enfermedad



- Si respuesta = 1
 - Si checkbox seleccionado
 - Nada
 - Si no
 - Sumar localizados con sin localizar
 - Almacenar parámetros para generador de gráficas
 - Terminar dialogo cargando datos
 - Recoger datos consulta2
 - GET a base de datos por enfermedad
 - Si respuesta=1
 - Si checkbox seleccionado
 - Nada
 - Si no
 - Sumar localizados con sin localizar
 - Almacenar parámetros para generador de gráficas
 - Terminar dialogo cargando datos
 - Recoger datos consulta2
 - GET a base de datos por enfermedad
 - Si respuesta=1
 - Si checkbox seleccionado
 - Nada
 - Si no
 - Sumar localizados con sin localizar
 -
 - Pasar datos a generador de gráficas
 - Terminar dialogo cargando datos
 - Mostrar gráfica
 - Si no
 - Terminar dialogo cargando datos
 - Mostrar mensaje no hay datos para esas fechas
- Si no
 - Terminar dialogo cargando datos
 - Mostrar mensaje no hay datos para esas fechas
- Consulta2
 - Ejecutar dialogo de cargando datos
 - Recoger datos usuario
 - Añadir datos a parámetros de consulta
 - GET a base de datos por país
 - Si respuesta = 1
 - Si checkbox seleccionado
 - Nada
 - Si no
 - Sumar localizados con sin localizar
 - Almacenar parámetros para generador de gráficas
 - Terminar dialogo cargando datos
 - Recoger datos consulta2



- GET a base de datos por país
- Si respuesta=1
 - Si checkbox seleccionado
 - Nada
 - Si no
 - Sumar localizados con sin localizar
 - Almacenar parámetros para generador de gráficas
 - Terminar dialogo cargando datos
 - Recoger datos consulta2
 - GET a base de datos por país
 - Si respuesta=1
 - Si checkbox seleccionado
 - Nada
 - Si no
 - Sumar localizados con sin localizar
 -
 - Pasar datos a generador de gráficas
 - Terminar dialogo cargando datos
 - Mostrar gráfica
 - Si no
 - Terminar dialogo cargando datos
 - Mostrar mensaje no hay datos para esas fechas
- Si no
 - Terminar dialogo cargando datos
 - Mostrar mensaje no hay datos para esas fechas

A continuación se detallan algunos de los términos que pueden resultar confusos para el lector

Modo: es una variable que va dentro del Intent generado por la actividad principal

Tipo de consulta: variable de tipo entero que puede valer 1 o 2 y sirve para determinar el tipo de consulta que se va a realizar.

Contenido lista: Se refiere al array de valores que se mostrará en la lista interactiva

Fecha1: fecha de inicio de la consulta

Fecha2: fecha de fin de la consulta

Mostrar mensaje: Se refiere a mostrar un mensaje flotante con distintos contenidos

Variable OK: variable que sirve para controlar que los parámetros introducidos son correctos

Actividad asíncrona: Android desde su versión 4 obliga a realizar todas las tareas pesadas mediante tareas asíncronas para no bloquear la interfaz del usuario

Dialogo cargando datos: dialogo que se muestra durante la petición de los datos al servidor

GET: llamada al fichero php para realizar la consulta en el servidor.

Consulta1: método al que se accede cuando el botón que se ha pulsado en la actividad principal es el de consulta por enfermedad

Consulta2: método al que se accede cuando el botón que se ha pulsado en la actividad principal es el de consulta por país

5.2.2 Layouts

Los layouts son archivos xml que configuran las pantallas que van a mostrarse. Cada uno es llamado por su actividad. En esta sección se muestran los layouts de cada actividad. Como se puede observar en los xml a continuación, en esta etapa se define toda la parte gráfica, desde las posiciones de los elementos hasta su color.

Activity_historial.xml

Este xml corresponde a la pantalla que se muestra al ejecutar la actividad Historial.java

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:paddingBottom="@dimen/activity_vertical_margin"
7     android:paddingLeft="@dimen/activity_horizontal_margin"
8     android:paddingRight="@dimen/activity_horizontal_margin"
9     android:paddingTop="@dimen/activity_vertical_margin"
10    android:background="@drawable/pruebafondo"
11
12    tools:context=".PorPaísesActivity" >
13
14    <Spinner
15        android:id="@+id/spinner1"
16        android:layout_width="98dp"
17        android:layout_height="wrap_content"
18        android:layout_x="20dp"
19        android:layout_y="58dp" />
20
21    <TextView
22        android:id="@+id/textView3"
23        android:layout_width="250dp"
24        android:layout_height="48dp"
25        android:layout_marginTop="48dp"
26        android:layout_x="0dp"
27        android:layout_y="5dp"
28        android:text="Paso1: Seleccione un país y una enfermedad"
29        android:textColor="#4099FF"
30        android:textSize="20sp" />
31
32    <Button
33        android:id="@+id/consulta"
34        android:layout_width="120dp"
35        android:layout_height="wrap_content"
36        android:layout_x="85dp"
37        android:layout_y="380dp"
38        android:text="Consultar"
39        android:textSize="18sp" />
40
41    <Spinner
42        android:id="@+id/Spinner01"
43        android:layout_width="98dp"
44        android:layout_height="wrap_content"
45        android:layout_x="20dp"
46        android:layout_y="100dp" />
47
48    <TextView
49        android:id="@+id/TextView1"
50        android:layout_width="250dp"
51        android:layout_height="48dp"
52        android:layout_marginTop="48dp"
53        android:layout_x="0dp"
54        android:layout_y="155dp"
55        android:text="Paso2: Seleccione un intervalo de fechas"
56        android:textColor="#4099FF"
57        android:textSize="20sp" />
```

Ilustración 50 Fichero XML 1


```
50     android:layout_height="48dp"
51     android:layout_marginTop="48dp"
52     android:layout_x="0dp"
53     android:layout_y="155dp"
54     android:text="Paso2: Seleccione un intervalo de fechas"
55     android:textColor="#4099FF"
56     android:textSize="20sp" />
57
58     <TextView
59         android:id="@+id/textView2"
60         android:layout_width="wrap_content"
61         android:layout_height="wrap_content"
62         android:layout_marginTop="15dp"
63         android:layout_x="15dp"
64         android:layout_y="220dp"
65         android:text="Seleccione fecha" />
66
67     <Button
68         android:id="@+id/button1"
69         android:layout_width="wrap_content"
70         android:layout_height="34dp"
71         android:layout_x="15dp"
72         android:layout_y="240dp"
73         android:background="#FFC4C4"
74         android:text="Cambiar fecha inicio"
75         android:textSize="16sp" />
76
77     <TextView
78         android:id="@+id/TextView01"
79         android:layout_width="wrap_content"
80         android:layout_height="wrap_content"
81         android:layout_marginTop="15dp"
82         android:layout_x="15dp"
83         android:layout_y="285dp"
84         android:text="Seleccione fecha" />
85
86     <Button
87         android:id="@+id/Button02"
88         android:layout_width="166dp"
89         android:layout_height="34dp"
90         android:layout_x="15dp"
91         android:layout_y="305dp"
92         android:background="#FFC4C4"
93         android:text="Cambiar fecha fin"
94         android:textSize="16sp" />
95
96     <CheckBox
97         android:id="@+id/checkbox1"
98         android:layout_width="wrap_content"
99         android:layout_height="wrap_content"
100         android:layout_x="15dp"
101         android:layout_y="340dp"
102         android:text="Utilizar solo tweets con localizacion" />
103
104 </AbsoluteLayout>
```

Ilustración 51 Fichero XML 2

Activity_info.xml

En las siguientes imágenes son del xml que corresponden a la pantalla que se muestra al ejecutar la actividad FAQ.java

Ilustración 52 Fichero XML 3

```

54     <TextView
55         android:id="@+id/textView5"
56         android:layout_width="280dp"
57         android:layout_height="45dp"
58         android:layout_x="16dp"
59         android:layout_y="298dp"
60         android:text="-POR ENFERMEDAD: te mostrará la proporción de afectados en cada país"
61         android:textSize="14sp" />
62
63     <TextView
64         android:id="@+id/textView1"
65         android:layout_width="280dp"
66         android:layout_height="67dp"
67         android:layout_x="16dp"
68         android:layout_y="240dp"
69         android:text="-HISTORIAL podrás observar la progresión de una enfermedad en un país"
70         android:textSize="14sp" />
71
72 </AbsoluteLayout>

```

Ilustración 53 Fichero XML 4

Activity_mail.xml

Este es el xml que corresponden a la pantalla que se muestra al ejecutar la actividad MainActivity.java

```

1  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:background="@drawable/pruebafondo"
6      android:paddingBottom="@dimen/activity_vertical_margin"
7      android:paddingLeft="@dimen/activity_horizontal_margin"
8      android:paddingRight="@dimen/activity_horizontal_margin"
9      android:paddingTop="@dimen/activity_vertical_margin"
10     tools:context=".MainActivity" >
11
12     <Button
13         android:id="@+id/buttonPais"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:layout_alignLeft="@+id/buttonEnfermedad"
17         android:layout_alignRight="@+id/buttonEnfermedad"
18         android:layout_below="@+id/buttonWhat"
19         android:layout_marginTop="46dp"
20         android:background="#FFC4C4C4"
21         android:text="Consulta por país"
22         android:textSize="24sp" />
23
24     <Button
25         android:id="@+id/buttonEnfermedad"
26         android:layout_width="wrap_content"
27         android:layout_height="wrap_content"
28         android:layout_alignLeft="@+id/historial"
29         android:layout_alignRight="@+id/historial"
30         android:layout_below="@+id/buttonPais"
31         android:layout_marginTop="44dp"
32         android:background="#FFC4C4C4"
33         android:text="Consulta por enfermedad"
34         android:textSize="24sp" />
35
36     <Button
37         android:id="@+id/historial"
38         android:layout_width="wrap_content"
39         android:layout_height="wrap_content"
40         android:layout_below="@+id/buttonEnfermedad"
41         android:layout_centerHorizontal="true"
42         android:layout_marginTop="52dp"
43         android:background="#FFC4C4C4"
44         android:text="Consultar historial País/Enfermedad"
45         android:textSize="24sp"
46         android:visibility="visible" />
47
48     <TextView
49         android:id="@+id/click"
50         android:layout_width="wrap_content"
51         android:layout_height="wrap_content"
52         android:layout_alignLeft="@+id/buttonPais"
53         android:layout_alignParentTop="true"
54         android:layout_marginLeft="30dp"
55         android:clickable="true"

```

Ilustración 54 Fichero XML 5

```

52         android:layout_alignLeft="@+id/button1"
53         android:layout_alignParentTop="true"
54         android:layout_marginLeft="30dp"
55         android:clickable="true"
56         android:onClick="onClick"
57         android:text="FA0"
58         android:textColor="#4099FF"
59         android:textColorHint="#4099FF"
60         android:textSize="15sp" />
61
62 </RelativeLayout>

```

Ilustración 55 Fichero XML 6

Activity_basicas.xml

En este xml se muestra la pantalla para la actividad ConsultasBásicas.java

```

1 <AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/and
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:background="@drawable/pruebafondo"
6   android:paddingBottom="@dimen/activity_vertical_margin"
7   android:paddingLeft="@dimen/activity_horizontal_margin"
8   android:paddingRight="@dimen/activity_horizontal_margin"
9   android:paddingTop="@dimen/activity_vertical_margin"
10  tools:context=".PorPaisesActivity" >
11
12  <Button
13      android:id="@+id/button2"
14      android:layout_width="88dp"
15      android:layout_height="wrap_content"
16      android:layout_x="100dp"
17      android:layout_y="365dp"
18      android:background="#FFC4C4C4"
19      android:text="Consultar"
20      android:textSize="18sp" />
21
22  <Button
23      android:id="@+id/button3"
24      style="?android:attr/buttonStyleSmall"
25      android:layout_width="wrap_content"
26      android:layout_height="wrap_content"
27      android:layout_marginTop="17dp"
28      android:layout_x="15dp"
29      android:layout_y="265dp"
30      android:background="#FFC4C4C4"
31      android:text="Seleccionar fecha fin"
32      android:textSize="16sp" />
33
34  <TextView
35      android:id="@+id/TextView1"
36      android:layout_width="wrap_content"
37      android:layout_height="wrap_content"
38      android:layout_marginTop="16dp"
39      android:layout_x="20dp"
40      android:layout_y="245dp"
41      android:text="Seleccione fecha" />
42
43  <Button
44      android:id="@+id/historial"
45      style="?android:attr/buttonStyleSmall"
46      android:layout_width="151dp"
47      android:layout_height="wrap_content"
48      android:layout_marginTop="18dp"
49      android:layout_x="15dp"
50      android:layout_y="185dp"
51      android:background="#FFC4C4C4"
52      android:text="Seleccionar fecha inicio"
53      android:textSize="16sp" />
54
55  <TextView

```

Ilustración 56 Fichero XML 7

```
56         android:id="@+id/textView2"
57         android:layout_width="wrap_content"
58         android:layout_height="wrap_content"
59         android:layout_marginTop="15dp"
60         android:layout_x="20dp"
61         android:layout_y="165dp"
62         android:text="Seleccione fecha" />
63
64     <TextView
65         android:id="@+id/textView1"
66         android:layout_width="wrap_content"
67         android:layout_height="wrap_content"
68         android:layout_marginTop="17dp"
69         android:layout_x="10dp"
70         android:layout_y="110dp"
71         android:text="Paso 2: elija un intervalo de tiempo"
72         android:textColor="#4099FF"
73         android:textColorLink="#4099FF"
74         android:textSize="20sp" />
75
76     <Spinner
77         android:id="@+id/spinner1"
78         android:layout_width="120dp"
79         android:layout_height="52dp"
80         android:layout_x="20dp"
81         android:layout_y="58dp" />
82
83     <TextView
84         android:id="@+id/textView3"
85         android:layout_width="250dp"
86         android:layout_height="48dp"
87         android:layout_marginTop="48dp"
88         android:layout_x="10dp"
89         android:layout_y="5dp"
90         android:text="Paso1: Seleccione un país de la lista"
91         android:textColor="#4099FF"
92         android:textSize="20sp" />
93
94     <CheckBox
95         android:id="@+id/checkbox1"
96         android:layout_width="wrap_content"
97         android:layout_height="wrap_content"
98         android:layout_x="15dp"
99         android:layout_y="320dp"
100        android:text="Utilizar solo tweets con localizacion" />
101
102 </AbsoluteLayout>
```

Ilustración 57 Fichero XML 8

5.3 Manifiesto

Es un archivo de configuración donde se pueden aplicar las configuraciones básicas de la aplicación. Su configuración puede realizarse a través de una interfaz gráfica a través de la sintaxis xml. El Android manifest está situado en la raíz de cada aplicación. Todas y cada una de las actividades de la aplicación deben ser declaradas en esta sección, sino no funcionarán. La actividad inicial de la aplicación debe ser declarada de tipo "LAUNCHER". A continuación se muestra el manifest de la aplicación.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.tweetfections"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="8"
9         android:targetSdkVersion="17" />
10    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
11    <application
12        android:allowBackup="true"
13        android:icon="@drawable/ic_launcher"
14        android:label="@string/app_name"
15        android:theme="@style/AppTheme" >
16        <activity
17            android:name="com.example.tweetfections.MainActivity"
18            android:label="@string/app_name" >
19            <intent-filter>
20                <action android:name="android.intent.action.MAIN" />
21
22                <category android:name="android.intent.category.LAUNCHER" />
23            </intent-filter>
24        </activity>
25        <activity
26            android:name="com.example.tweetfections.Historial"
27            android:label="@string/title_activity_historial" >
28        </activity>
29        <activity
30            android:name="com.example.tweetfections.FAQ"
31            android:label="@string/title_activity_info" >
32        </activity>
33        <activity
34            android:name="com.example.tweetfections.ConsultasBasicas"
35            android:label="@string/title_activity_por_paises" >
36        </activity>
37        <activity android:name="org.achartengine.GraphicalActivity" />
38
39    </application>
40
41 </manifest>
```

Ilustración 58 Android manifest

6 Evaluación de resultados

En esta sección se muestran los resultados obtenidos de varias ejecuciones en distintas partes del programa. Por un lado se encuentran las ejecuciones por el programa en Java y por otro lado las ejecuciones realizadas por la APP y en último lugar, aunque no son tiempos de ejecuciones se incluye un estudio de los ficheros con datos recogidos de Twitter. Además se mostrarán también una serie de tablas con los datos que han servido para realizar los coeficientes de ajuste y de reparto para los tweets con localización

Las pruebas se han realizado en un servidor y red local por lo que los resultados seguramente serían peores si las pruebas se realizasen en condiciones normales. Los recursos son los mencionados en el apartado equipos. Para el cálculo de tiempos se ha utilizado la función de Java `System.currentTimeMillis()` que devuelve el tiempo en milisegundos. Por cada prueba se han realizado 10 pruebas entre las que se ha obtenido la media, el máximo y el mínimo.

6.1 Tiempo de ejecución del programa en Java

Este es el tiempo que tarda en ejecutarse el programa. El programa se ejecuta una vez al día y, debido a las mencionadas limitaciones de la API de Twitter, tras realizarse cada consulta de una enfermedad debe esperarse una hora hasta realizar la siguiente. Por lo que como hay 5 consultas de partida tenemos un tiempo de espera de 4 horas (después de la 5ª consulta no se espera). Además a este tiempo hay que sumarle el tiempo que tarda el programa en recibir todos los datos de Twitter, que para cada enfermedad es distinto porque el volumen es distinto. Y por último hay que añadir el tiempo que tarda el programa en preparar las inserciones y enviarlas.

En estas tablas se muestra el número de tweets recogidos por cada expresión. Y el tiempo empleado en su recogida y su escritura en fichero.

Tabla 109 Resultados 1

Expresión	Mínimos resultados obtenidos	Máximos resultados obtenidos	Media de resultados obtenidos	Tiempo medio de recopilación de resultados (ms)
Tengo gripe	229	443	336	3494,4
Estoy con gripe	102	305	255	2652
Tengo gripa	234	307	271	2818,4
Estoy con gripa	337	428	383	3983,2
He pillado la gripe	95	201	148	1539,2
Tengo alergia	330	430	380	3952
Estoy con alergia	123	322	223	2319,2
Tengo asma	65	243	154	1601,6
Respirar asma	230	403	317	3296,8
Tengo gastroenteritis	150	372	261	2714,4

Estoy con gastroenteritis	201	405	303	3151,2
Tengo diarrea	188	321	255	2652
Tengo vómitos	192	296	244	2537,6
Estoy enfermo con vómitos	79	165	122	1268,8
Me duele la garganta	154	411	283	2943,2
Tengo faringitis	137	258	198	2059,2
Me duelen las anginas	265	482	374	
Tengo anginas	198	447	323	3359,2
Estoy enfermo con anginas	154	302	228	2371,2
Estoy constipado	332	481	407	4232,8
Estoy acatarrado	234	389	312	3244,8
Tengo catarro	146	340	243	2527,2
Estoy resfriado	173	297	235	2444
He pillado un catarro	91	230	161	1674,4

En la siguiente tabla se muestra un resumen de los tiempos para la ejecución de cada consulta a Twitter por cada enfermedad (suma de los tiempos de cada expresión)

Tabla 110 Resultados 2

Enfermedad consultada	Tiempo para la consulta a Twitter (ms)
Gripe	14487,2
Alergia	11169,6
Gastroenteritis	12324
Catarro	10732,8
Anginas	14123,2

Además de las consultas se mide también el tiempo que se tarda en hacer las búsquedas en los ficheros y preparar las inserciones en la base de datos.

Tabla 111 Resultados 3

Fichero accedido	Tiempo medio para acceder leer y distribuir datos
Gripe	50ms
Alergia	35ms
Gastroenteritis	38ms
Catarro	32ms
Anginas	48ms



Como estas búsquedas se realizan 2 veces por país y 9 veces por fichero, en el cómputo global se incluirán multiplicadas por 18.

Y por último las inserciones en la base de datos. Como las inserciones se realizan a la vez, sólo hay una tabla que comprende el tiempo para realizar una petición y el tiempo para la inserción. En total son 9 consultas de registro y 9 inserciones (dos por cada país). Tras 10 ejecuciones de este proceso se han obtenido los siguientes datos. 3654+62836,6

Tabla 112 Resultados 4

Proceso	Tiempo mínimo	Tiempo máximo	Tiempo medio
Consultas e inserciones en base de datos.	20034,7	27433,23	23722,965

Todo lo anterior sumado más el tiempo de espera para cada consulta suma un total medio de 14490213,75 milisegundos. De los cuales 14400000 milisegundos (4 horas) corresponden al tiempo de espera entre consulta y consulta y el tiempo efectivo de trabajo del programa es una media de 90213,765 es decir aproximadamente 1 minuto y medio de ejecución efectiva para consultar a Twitter crear los ficheros e insertar en la base de datos calculados.

6.2 Consultas desde la aplicación móvil.

Hay que mencionar que la ejecución de todas estas pruebas se han realizado a nivel local por lo que estos datos son sólo orientativos y deberían realizarse pruebas utilizando 3G y WI-FI para obtener unos resultados mejores y más realistas.

Consultas por país

Las pruebas de consulta por país se realizan 10 veces variando tanto fechas como países utilizados para la consulta.

Tabla 113 Resultados 5

Proceso	Tiempo mínimo	Tiempo máximo	Tiempo medio
Consulta por país	2387,5	3654,3	3020,9

Consultas por enfermedad

Este tipo de consulta es más costosa que la anterior porque en vez de acceder a un único registro por cada fecha a consultar se accede a tantos como países hay en el sistema. Aunque esto no se traduce en peticiones SQL si se traduce en trabajo en la base de datos.

Tabla 114 Resultados 6

Proceso	Tiempo mínimo	Tiempo máximo	Tiempo medio
Consulta por enfermedad	4577,5	6222,3	5399,9



Consultas de historial

Por último están las consultas por historial. En este proceso solo se envía una petición MYSQL pero el tiempo que tarda en ejecutarse depende del intervalo de la fecha dado que accede a todos los registros comprendidos entre la fecha de inicio y la fecha de fin de la consulta. Por esta razón se han incluido distintos intervalos de fechas para el análisis del tiempo para estas consultas.

Tabla 115 Resultados 7

Proceso	Tiempo mínimo	Tiempo máximo	Tiempo medio
Consulta de historial(intervalo de 5 días)	2721	3652,2	3186,6
Consulta de historial(intervalo de 10 días)	3112,8	4834,87	3973,835
Consulta de historial(intervalo de 30 días)	5021,33	8032,1	6526,715

Como puede observarse los tiempos de ejecución crecen en función del intervalo a consultar. Por lo tanto habría que controlar el tiempo que se puede tardar para realizar esta consulta o el máximo intervalo que se puede seleccionar en el sistema.

Estas consideraciones, como se han conocido una vez casi terminado el proyecto, se han incluido en la sección “Alternativas de diseño”.

6.3 Coeficientes de ajuste y reparto

En esta sección se explica cómo se han obtenido los coeficientes que se aplica para el reparto de tweets y para el ajuste del volumen.

Esta tabla muestra los tweets con localización válida, los tweets con localizaciones inválidas y los tweets sin localización recogidas de 10 consultas por enfermedad distintas. Los campos de la tabla son tweets con localización válida (tweets CLV), tweets con localización inválida (tweets CLI), tweets sin localización (tweets SL) y las comparaciones y porcentaje entre ellos.

Tabla 116 Resultados 8

Consulta	Tweets CLV	Tweets CLI	Tweets SL	Tweets CL vs Tweets SLV	Porcentaje CLV
1	234	332	187	234 vs 519	31%
2	320	540	304	320 vs 844	27,5%
3	50	260	114	50 vs 374	11,8%
4	27	342	89	27 vs 431	6%

5	75	328	132	75 vs 460	14%
6	345	544	288	345 vs 832	29,3%
7	107	569	201	107 vs 770	12,2%
8	342	532	233	342 vs 765	30,8%
9	221	324	157	221 vs 481	31,4%
10	68	234	69	68 vs 303	18%
				Media	21,2%

La tabla anterior sirve para hacerse una idea de qué porcentaje de tweets en el sistema contienen una localización real y qué porcentaje contiene una localización adjudicada por el programa. Como puede verse sólo el 21,2% de los tweets recogidos contienen una localización válida para el sistema. Este porcentaje podría aumentar en gran medida si se contemplasen nombres de ciudades y pueblos a buscar en los ficheros de localizaciones.

En la siguiente tabla se muestran los países que del sistema, su población, el ranking en Twitter comparados entre ellos y el porcentaje de reparto adjudicado.

Tabla 117 Resultados 9

País	Población (en millones de personas)	Ranking en Twitter (entre estos países)	Porcentaje de reparto
España	47,27	2	17%
México	112,3	1	20%
Argentina	41,09	5	12%
Venezuela	29,9	3	15%
Colombia	47,7	4	14%
Perú	29,99	7	6%
Chile	17,47	6	6%
Ecuador	15,49	8	4%
Guatemala	15,08	9	4%

Como puede observarse los coeficientes se han repartido pensando más en su ranking según su número de cuentas y uso de Twitter que en su población, ya que al parecer, la población no es un factor determinante en el uso de Twitter. Por ejemplo España se encuentran muy por delante que Argentina o Colombia en el uso de Twitter a pesar de tener una población parecida.



Por último en esta tabla se muestra el volumen de 10 consultas del sistema y el análisis del contenido de los tweets. Los campos de la tabla son tweets recogidos, tweets dentro del contexto (tweets DDC), tweets fuera del contexto (tweets FDC) y el porcentaje que supone estos dos últimos campos en el total.

Tabla 118 Resultados 10

Consulta	Tweets DDC	Tweets FDC	Porcentaje FDC
1	543	43	7,3%
2	347	41	10,5%
3	674	101	13,03%
4	780	92	10,5%
5	361	93	20,5%
6	466	78	14,3%
7	489	84	14,65%
8	665	102	13,29%
9	710	60	7,8%
10	693	32	4,41%
		Media	11,7%

Como muestra la tabla anterior el porcentaje de tweets fuera de contexto para las expresiones buscadas es de 11,7% una cifra muy cercana al coeficiente utilizado en el sistema: 12%.



7 Presupuesto

En esta sección se detalla un desglose aproximado de los costes de desarrollo del proyecto.

7.1 Costes

En la siguiente tabla se incluyen los costes de los distintos elementos hardware que forman el sistema

Tabla 119 Costes 1

Hardware	Precio	Unidades	Total
Ordenador personal para el desarrollo	920,00 €	1	920 €
Smartphone Samsung Galaxy S II para realizar pruebas.	255,00 €	1	255 €

En esta tabla puede observarse el coste de los elementos de software que utiliza el sistema.

Tabla 120 Costes 2

Software	Precio	Unidades	Total
Licencia de sistema operativo Windows 7	120 €	1	164 €
Licencia Microsoft Office Hogar y Estudiantes 2013	119 €	1	199 €
WAMP	0 €	1	0 €
Software libre	0 €	1	0 €

Cabe mencionar que en la tabla anterior, si el servidor Apache o la base de datos MySQL fuesen a tener un uso corporativo habría que pagar una licencia.

Los costes de hardware y software anteriores, se ajustan en función de la amortización. Se considera que el tiempo de vida del hardware (y software instalado) citado es de 36 meses, y la duración del proyecto, mostrada en el diagrama de planificación, es de 6 meses.

Tabla 121 Costes

Elemento	Coste mensual	Duración del proyecto	Coste amortizado
Hardware	32,65 €	6	261,2 €
Software	10,09 €	6	60,54 €
Total			321,74 €

Además de los componentes del sistema hay que incluir el factor humano que corresponde a las horas trabajadas. Aunque este proyecto lo ha realizado una sola persona, a continuación se detallan los perfiles de los trabajadores necesarios, su precio por hora de trabajo y un diagrama con las distintas fases y sus horas necesarias para realizar el proyecto en un entorno laboral.

A continuación se expone un diagrama con las distintas fases de desarrollo del proyecto y las horas necesarias para desarrollar cada fase.

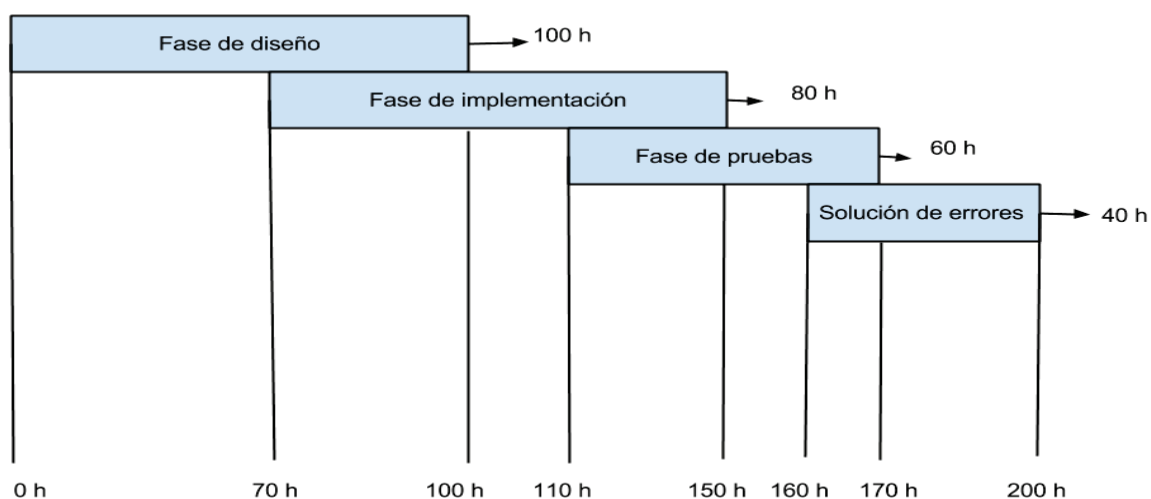


Ilustración 59 Diagrama de etapas



Dichas fases son llevadas a cabo por cada uno de los integrantes del equipo y con una dedicación de tiempo similar a la que en la imagen se muestra. Por lo tanto el coste del personal de trabajo quedaría de la siguiente forma:

Tabla 122 Costes

Trabajador	Precio hora	Horas necesarias	Total
Becario diseñador	5 €	200	1000 €
Programador en Java	10 €	200	2000 €
Programador en Android	10 €	200	2000 €
Programador Apache/MySQL	12 €	200	2400 €

Según los datos calculados en las tablas anteriores el coste final del proyecto sería:

Tabla 123 Costes 4

Parte del sistema	Total
Hardware	261 €
Software	60,54 €
Personal	7400 €
Total	7721 €



8 Conclusiones y líneas futuras

Este Trabajo de Fin de Grado ha alcanzado la mayoría de objetivos marcados. Se ha conseguido implementar un servidor en un ordenador personal, una aplicación Android y un programa en Java que busca en Twitter, todo ello de manera satisfactoria. El proyecto inicial sufrió un cambio de dirección cuando Twitter actualizó la política y funcionalidades de su API pero el resultado ha sido óptimo.

Gracias al sistema se puede tener acceso a información sobre las enfermedades en distintos países proporcionando de manera fácil rápida y sencilla al usuario una información que en Internet no es tan accesible como pudiera parecer. Gracias de esto, las personas que tengan curiosidad, o planeen un viaje a alguno de los países contemplados en el sistema, podrán planificar mejor esos viajes o adquirir el conocimiento que buscan respecto al tema.

También es verdad que muchos de los datos obtenidos y mostrados en la aplicación no se ajustan completamente a la realidad debido a que se han realizado ponderaciones y ajustes porcentuales. Pero el resultado final, comparando los datos mostrados, con los datos analizados a mano en Twitter, se llega a la conclusión de que para el tiempo y los recursos disponibles, así como para el ámbito del proyecto, el ajuste es más que suficiente.

De cara a mejorar la aplicación se podría seguir las siguientes líneas futuras:

- Aumentar los registros de localizaciones del sistema a ciudades y pueblos para conseguir un índice mayor de coincidencia y así poder ofrecer resultados más ajustados.
- Incluir el alquiler de robots de localización de tweets para ajustar aún más los resultados.
- Incluir un mapa gráfico interactivo para hacer más sencillo y visual el acceso a los resultados.
- Desarrollar la aplicación en Windows Mobile y en iOS para conseguir un mayor número de usuarios.
- Incluir en la aplicación móvil información sobre la prevención de las enfermedades del sistema.
- Añadir más idiomas de búsqueda y por ende más países al sistema.
- Añadir más enfermedades al sistema y la aplicación.
- Ponerse en contacto con Twitter para que proporcione un mayor número de consultas al programa en Java y conseguir así reducir el tiempo de ejecución del programa.
- Añadir más expresiones de búsqueda por cada enfermedad
- Incluir un analizador de contexto automático para ajustar mejor los coeficientes de tweets válidos y tweets inválidos.
-

Para terminar esta sección resta mencionar que, durante la realización del proyecto, se han afianzado conocimientos adquiridos a lo largo de toda la carrera. Por un lado se han mejorado las habilidades de programación en Java Android y MySQL y por otro se ha conseguido una información muy útil sobre el funcionamiento de los sistemas distribuidos implementados de manera real y que pretenden conseguir un objetivo común.

Además la resolución de errores, el rediseño y el trabajo por fases han brindado al alumno una capacidad resolutoria e imaginativa mayor y mejor que la que tenía antes de la realización del trabajo.



9 Bibliografía

1. **Cooper Bowen, Rcihard**; A. L. Coar, Ken. *Apache práctico*. España: Anaya Multimedia-Anaya interactiva, 2004. 288 p. ISBN: 8441517045
2. **Eckel, Bruce**. *Thinking in Java*. 4ª ed. Prentice-Hall. 1500 p. ISBN: 0131872486
3. **Verodmar**. *Guía de instalación de WampServer* [En línea]
<http://www.slideshare.net/verodmar/wampserver-gua-de-instalacin>
4. **Gironés, Jesus Tomás**. *El gran libro de Android*. España: Marcombo S.A., 2013. 446 p. ISBN: 8426719767.
5. **Vaswan Vickram**. *Fundamentos de PHP*. España: McGraw-Hill Interamericana de España S.L., 2011. 480 p. ISBN: 9701071328.
6. **W3schools.com** *Json info*. [En línea]. <http://www.w3schools.com/json/default.asp>
7. **W3schools.com** *Jquery info*. [En línea] <http://www.w3schools.com/jquery/default.asp>
8. **W3schools.com** *PHP info*. [En línea] <http://www.w3schools.com/php/default.asp>
9. **Android Developers**, *Android Calendars*
<http://developer.android.com/reference/java/util/Calendar.html>
10. **Android Developer**, *Android CheckBox*
<http://developer.android.com/reference/android/widget/CheckBox.html>
11. **Tamada, Ravi**. *How to connect Android with PHP, MySQL*. [En línea]
<http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/>
12. **Alonsojpd**. *Acceso a MySQL desde Android con Eclipse, Java y JDBC*. [En línea]
<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=656>
13. **Ayan, Omer**. *Servidor web con php y mysql en android*. [En línea]
<http://openwebcms.es/2013/servidor-web-con-php-y-mysql-en-android/>



14. **Cipolat, Sebastian.** *Login en Android usando php y mysql* [En línea]
<http://androideity.com/2012/07/05/login-en-android-usando-php-y-mysql/>
15. **Castromil, Juan Carlos.** *Ranking de países más Twitter Adictos.* [en línea]
<http://blogs.20minutos.es/clipset/ranking-de-paises-mas-twitter-adictos/>
16. **Twitter4j.** *Interface Twitter* [En línea] <http://twitter4j.org/javadoc/twitter4j/Twitter.html>
17. **Google Developers.** *Achartengine.* [En línea] <https://code.google.com/p/achartengine/>
18. **Achartengine.org.** *Achartengine javadoc* [En línea]
<http://www.achartengine.org/content/javadoc/index.html>
19. **Moure Conde, Andrés.** *Cómo usar PHP para conectar Android.* [En línea]
<http://nosoandroid.blogspot.com.es/2013/03/como-usar-php-para-conectar-android.html>
20. **Kumar Saurabh.** *Run Apache, MySQL, PHP – Web server on Android mobile or Tablet* [En línea] <http://techotv.com/run-apache-mysql-php-http-web-server-android-os-phone-tablet/>
21. **Chuwiki.chuidiang.org.** *Lectura y Escritura de Ficheros en Java* [En línea]
http://chuwiki.chuidiang.org/index.php?title=Lectura_y_Escritura_de_Ficheros_en_Java
22. **Wikipedia.org.** *List of countries by population* [En línea]
http://en.wikipedia.org/wiki/List_of_countries_by_population